
효율적인 이벤트 검색을 위한 프로파일 기반 질의 처리 방법

김창훈* · 김태영* · 김종민* · 반재훈* · 김동현**

*고신대학교 인터넷비즈니스학과, **동서대학교 컴퓨터정보공학부

Design of Query Processing based on Profiles for Efficient Searching Events

ChangHoon Kim* · TaeYoung Kim* · JongMin Kim* · ChaeHoon Ban* · DongHyun Kim**

*Dept. of Internet Business, Kosin University

**Division of Computer & Information Engineering, Dongseo University

E-mail : {chkim, tykim, jmkim, chban}@kosin.ac.kr* · pusrover@dongseo.ac.kr**

요 약

최근 다양한 정보검색 기법의 발달로 인해 사용자는 인터넷으로부터 유용한 데이터를 쉽게 획득할 수 있다. 이러한 데이터는 스트림 데이터와 유사하게 연속적으로 발생하기 때문에 인터넷에서 발생하는 대량의 데이터 중에서 사용자의 요구에 맞는 데이터만을 추출하기 위한 기법이 필요하다. 전통적인 데이터 추출 방법에서는 데이터를 먼저 데이터베이스에 저장한 후에 사용자의 요구를 질의로 변환하여 처리하였다. 그러나 이러한 방법은 연속적으로 발생하는 대용량의 데이터에 대하여 사용자 질의를 처리하는 경우에 비효율적인 문제가 있다.

이 논문에서는 인터넷 환경에서 대용량의 연속적인 데이터 스트림으로부터 사용자의 요구에 맞는 데이터를 추출하는 질의처리 기법을 제안한다. 먼저 인터넷에서 발생하는 데이터의 변화를 이벤트로 표현하고, 사용자의 요구사항을 프로파일로 정의한 이벤트-프로파일 모델을 제안한다. 그리고 이 모델을 기반으로 이벤트와 프로파일을 효율적으로 처리하기 위한 여과방법을 제안한다.

ABSTRACT

Recently, it is possible for users to acquire necessary data easily as the various schemes of the searching information are developed. Since these data rise continuously like stream data, it is required to extract the appropriate data for the user's needs from the mass data on the internet. In the traditional scheme, they are acquired by processing the user queries after the occurred data are stored at a database. However, it is inefficient to process the user queries over the large volume of continuous data by using the traditional scheme.

In this paper, we propose the query processing scheme to extract the data efficiently for the user requirements from the large volume of continuous data. On the proposed scheme, we present the Event-Profile Model to define the data occurrence on the internet as the events and the user's requirements as the profiles. We also show the filtering scheme to process the events and the profiles efficiently.

키워드

Event, Profile, Event-Action, Filtering, Query Processing

1. 서 론

정보검색 기법의 발달로 인해 사용자는 인터넷으로부터 유용한 데이터를 쉽게 획득할 수 있다. 그러나 사용자는 자신이 원하는 데이터를 얻기 위하여 많은 시간을 할애해 검색을 수행해야 한다. 따라서 사용자가 원하는 데이터를 자동으로

추출하고 통지하는 시스템이 필요하다.

전통적인 데이터 추출 방법에서는 데이터를 먼저 데이터베이스에 저장한 후에 사용자의 요구를 질의로 변환하여 처리하였다. 그러나 이러한 방법은 연속적으로 발생하는 대용량의 데이터에 대하여 사용자 질의를 매번 처리하므로 비효율적인

문제가 있다.

이 논문에서는 효율적인 데이터 검색을 위한 질의 처리 방법에 대하여 제안한다. 먼저 데이터베이스에 개체가 삽입, 삭제, 변경 될 때의 상태 변화를 이벤트로, 특정 개체에 대한 사용자의 질의를 프로파일로 정의한 이벤트-프로파일 모델을 제시한다. 그리고 효율적인 이벤트 검색을 위한 이벤트-프로파일 처리 시스템을 제안하고 2단계로 처리하는 이벤트-프로파일 여과방법에 대하여 기술한다.

이 논문의 구성은 다음과 같다. 먼저 2장에서는 관련 연구에 대하여 소개한다. 3장에서는 논문의 연구 동기와 문제를 정의하고 이벤트와 프로파일을 정의한다. 4장에서는 이 논문에서 제안한 이벤트-프로파일 처리 시스템에 대하여 기술하고 효율적인 검색을 위한 이벤트-프로파일 여과 방법에 대하여 제안한다. 마지막으로 5장에서는 결론 및 향후 연구에 대해서 기술한다.

II. 관련 연구

최근 많은 응용분야에서 스트림 데이터가 발생하고 있다. 연속해서 지속적으로 발생하는 스트림 데이터를 처리하기 위해서는 순간 질의가 아닌 연속질의가 필요하며, 슬라이딩 윈도우에 대한 질의를 수행 할 수 있어야 한다[1][2].

기존의 관계형 DBMS에서는 성능의 한계로 인해 이러한 스트림데이터의 처리에 어려움이 있었으나, 고성능 메모리 중심, DBMS의 등장으로 인해 충분한 처리능력을 갖추게 되었다. 이에 따라 트리거를 이용하여 관계형 DBMS에서 연속 질의 지원을 고려할 수 있다. [4]에서는 고성능 RDBMS의 높은 삽입과 갱신 성능을 가정하여, 슬라이딩 윈도우에 대한 연속 질의를 트리거를 이용하여 지원하는 방법을 제시하였다. [5]에서는 고성능 메인메모리 데이터베이스에서 보다 효율적인 트리거를 통한 슬라이딩 윈도우의 지원방법을 제시하였다.

이 논문에서는 이러한 관련 연구를 바탕으로 이벤트와 프로파일을 BNF(Backus-Naur Form)로 표현한 이벤트-프로파일 모델을 제시한다. 그리고 효율적인 이벤트 검색을 위한 이벤트-프로파일 처리 시스템을 제안하고 2단계로 처리하는 이벤트-프로파일 여과방법에 대하여 기술한다.

III. 문제 정의

3.1 자동검색통지 시스템

인터넷으로부터 사용자는 유용한 데이터를 쉽게 획득할 수 있지만 정보의 양이 방대하여 사용자가 원하는 데이터를 자동으로 추출하고 통지하는 시스템이 필요하다. 기존의 방법에서는 데이터

를 먼저 데이터베이스에 저장한 후에 사용자의 요구를 질의로 변환하여 처리하였다. 그러나 이러한 방법은 연속적으로 발생하는 대용량의 데이터에 대하여 사용자 질의를 매번 처리하므로 비효율적인 문제가 있다.

예를 들어 그림 1과 같은 게시판 자동검색통지 시스템이 있다고 가정하자. 이 시스템은 사용자의 요구사항을 질의로 저장하고 게시판을 검색하여 통지하는 시스템이다. 이 시스템은 새로운 데이터가 삽입될 때 마다 저장되어 있는 다수의 사용자의 질의를 매번 수행하는 문제가 발생한다. 본 논문에서는 이러한 문제를 해결하기 위해 이벤트와 프로파일을 여과하여 처리하는 방법을 제시한다.

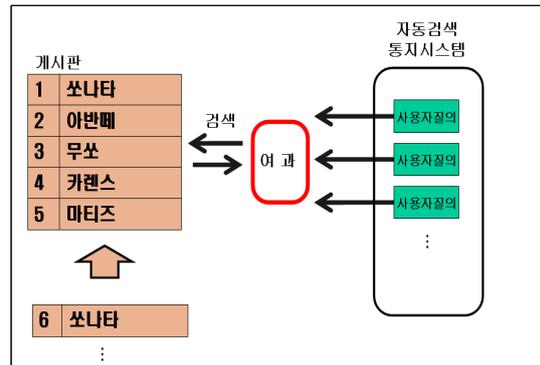


그림 1. 여과를 이용한 자동검색통지 시스템

3.2 이벤트와 프로파일

사용자는 특정 객체에 대해 어떤 변화가 발생하는가에 관심을 가진다. 객체에 대한 상태의 변화를 이벤트로 정의한다[3]. 이벤트의 실제 발생을 이벤트 인스턴스 e 라 하고, 발생시간을 t 라 할 때, 하나의 이벤트 e 는 $t(e)$ 로 나타내고 이벤트 집합 E 는 (식 1)과 같이 정의한다.

$$E = \{ t_1(e), t_2(e), \dots, t_n(e) \} \dots \dots \dots \text{(식 1)}$$

프로파일이란 사용자에 의해 기술된 객체에 대한 관심으로 정의한다. 사용자의 의해 정의된 객체에 대한 관심을 프로파일 p 라고 할 때, 프로파일 집합 P 는 (식 2)와 같이 정의 된다.

$$P = \{ p_1, p_2, \dots, p_n \} \dots \dots \dots \text{(식 2)}$$

이 논문에서는 데이터베이스에 개체가 삽입, 삭제, 변경 될 때의 상태 변화를 이벤트로 정의하고 특정 개체에 대한 사용자의 질의를 프로파일로 정의한다. 이때 이벤트는 스트림데이터와 같이 시간에 연속적으로 발생하는 데이터이다. 또한 프로파일은 순간질의가 아닌 계속해서 처리해야하는 연속질의의 형태이다. 또한 이벤트 발생 시 효율적으로 프로파일을 처리하기 위하여 이벤트와 프로파일을 각각 BNF(Backus-Naur Form)로 표현하고 이벤트와 프로파일을 여과하여 검색하는 방법에 대하여 제시한다.

IV. 이벤트-프로파일 처리 시스템

4.1 이벤트-프로파일 모델

이 논문에서는 데이터베이스에 개체가 삽입, 변경 될 때의 상태 변화를 이벤트로 정의하고 그림 2와 같이 BNF로 표현한다. 이벤트는 Type, TimeStamp, KeywordList의 속성으로 정의된다. Type은 Insert, Update의 속성 값을 가지며, TimeStamp는 이벤트의 발생시간을 나타내는 속성 값으로 여과단계에서 사용된다. 또한 KeywordList에는 여러 개의 키워드 값을 가질 수 있다.

```
Event ::= <Type><TimeStamp><KeywordList>
Type ::= 'Insert' | 'Update'
TimeStamp ::= <StringLiteral>
KeywordList ::= <StringLiteral> { <StringLiteral> }
```

그림 2. BNF로 표현한 이벤트

<그림 3>은 이벤트의 BNF 예제이다. 데이터베이스에 개체가 삽입 되었을 때 이벤트가 발생하며 이 정보를 BNF형식으로 변환하여 저장한다.

```
마티즈 600만원에 팝니다. 4년 되었으며 옵션은 기본옵션입니다.
Type = 'Insert'
KeywordList = '마티즈', '600만원', '4년', '기본옵션'
```

그림 3. BNF로 표현한 이벤트 예

```
Profile ::= 'Event' <EventType>
          'Condition' <ConditionExpr>
          'Action' <ActionList>
EventType ::= 'Insert' | 'Update' | 'Insert&Update'
ConditionExpr ::= <Time><SelectList>
Time ::= <PeriodTime><MaintenanceTime><TimeStamp>
PeriodTime ::= <StringLiteral> | 'All'
MaintenanceTime ::= <StringLiteral> | 'All'
TimeStamp ::= <StringLiteral>
SelectList ::= <Select> { <LogicalOP> <Select>}
Select ::= <ArithmeticOP> <StringLiteral>
LogicalOP ::= 'AND', 'OR'
ArithmeticOP ::= '<', '>', '>=', '<=', '<='
ActionList ::= <Protocol>, <ID>, <Message>
Protocol ::= 'SMS' | 'E-Mail' | 'Etc'
ID ::= <StringLiteral>
Message ::= <StringLiteral>
```

그림 4. BNF로 표현한 ECA기반 프로파일

이 논문에서는 특정 개체에 대한 사용자의 질의를 프로파일로 정의하고 그림 4와 같이

ECA(Event-Condition-Action)를 기반으로 하여 BNF로 표현한다. 프로파일은 EventType, ConditionExpr, ActionList로 정의되며 EventType은 어떠한 이벤트에 대하여 관심이 있는냐에 따라 Insert, Update, Insert&Update의 속성 값을 가질 수 있다.

또한 ConditionExpr은 Time, SelectList로 정의되며 Time은 프로파일의 검색주기시간인 PeriodTime, 프로파일 유지시간인 MaintenanceTime, 그리고 프로파일 발생시간 등 시스템에서 사용되는 TimeStamp의 속성을 가진다. SelectList는 사용자의 질의를 표현하는 속성으로 Select, LogicalOP의 속성을 가진다. 사용자가 정의한 이벤트 발생시 통지하는 정보를 나타내는 ActionList는 Protocol, ID, Message의 속성을 가진다. 그림 5는 BNF로 표현된 프로파일의 예이다.

```
가격이 600만원인 소나타를 2분 간격으로 검색하여 SMS 문자로 통지해 주시오.
EventType = 'Insert&Update'
PeriodTime = '2min'
MaintenanceTime = 'All'
SelectionList = '<', '소나타', 'AND', '>', '600만원'
Protocol = 'SMS'
ID = '01038734563'
Message = '...'
```

그림 5. BNF로 표현한 프로파일의 예

4.2 시스템 구성도

그림 6은 전체 시스템 구성도이다. 먼저 사용자는 프로파일관리자를 통해 프로파일을 정의하며 정의된 프로파일은 프로파일저장소에 저장된다. 새로운 개체가 데이터베이스에 삽입 또는 변경되면 이벤트가 발생하고 이벤트는 파서를 통해 추출되어 이벤트저장소에 저장된다. 그리고 이벤트가 발생되었을 때 저장된 프로파일과 이벤트를 여과하여 후보를 줄이고 최종적으로 정제 단계를 거쳐 사용자에게 문자, e-mail 등을 통해 통지한다.

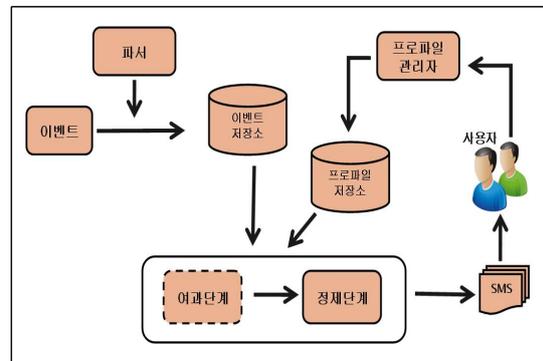


그림 6. 이벤트-프로파일 시스템 구성도

그림 7은 이 시스템에서 가장 중요한 여과를 단계별로 표현한 그림이다. 여과는 크게 프로파일 여과와 이벤트여과로 나누어진다. 먼저 프로파일 여과에서는 관심 있는 이벤트의 타입(EventType Step), 프로파일 검색유지시간(MaintenanceTime Step), 그리고 프로파일 검색주기시간(PeriodTime Step)을 고려하여 여과를 수행한다. 프로파일여과를 수행하게 되면 최종 정제 단계에 필요한 프로파일만을 추출하게 된다.

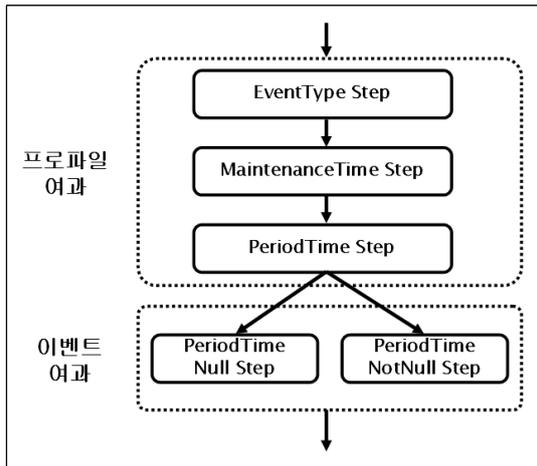


그림 7. 이벤트-프로파일 여과

이벤트여과에서는 검색주기시간에 따라 2가지 경우로 나누어 수행된다. 검색주기 시간이 없는 프로파일의 경우에는 이벤트가 발생할 때 마다 검색되어야하므로 이 경우는 새로 발생한 이벤트만이 여과된다. 그러나 검색주기시간이 있는 프로파일의 경우에는 시간을 고려하여 검색대상이 되어야하는 이벤트를 찾아 여과한다.

```

    Procedure PeriodTimeNotNullStep(Events EventSet, Profiles ProfileResultSet)
    begin
    for each object P in ProfileResultSet
    do
        for each object E in EventSet
        do
            if(CurrentTime - P.PeriodTime <= E.InsertTime)
            Add E to EventResultSet
            endfor
            RefinementStep(EventResultSet, P)
            EventResultSet = Null
        endfor
    end
    end
    
```

그림 8. PeriodTimeNotNullStep 알고리즘

그림 8은 이벤트여과 중에서 프로파일에 검색 주기가 있는 경우에 해당되는 이벤트를 여과하는 PeriodTimeNotNullStep 알고리즘이다. 이 단계에

서는 이미 여과된 각각의 프로파일들에 대하여 정제를 수행해야 될 이벤트를 여과한다. 각 프로파일에 대하여 현재시간에서 검색주기시간의 차보다 큰 시간을 타임스탬프로 가진 이벤트들을 추출한다. 왜냐하면 이전 검색 후에 발생한 이벤트이므로 검색대상이 되기 때문이다.

이미 여과된 각 프로파일에 대하여 해당되는 이벤트를 여과한 후에 최종 정제를 수행하게 된다. 정제는 인덱스를 사용하여 수행하며 정제 후에 최종으로 사용자에게 등록된 메시지를 통보한다. 이 논문에서는 정제와 통보는 고려하지 않는다.

V. 결론 및 향후 연구

인터넷으로부터 발생하는 대량의 데이터 중에서 사용자의 요구에 맞는 데이터만을 자동으로 추출하고 통지하는 기법이 필요한데 이 경우에 연속적으로 발생하는 대용량의 데이터에 대하여 사용자 질의를 매번 처리하므로 비효율적인 문제가 있다. 이 논문에서는 효율적인 데이터 검색을 위한 질의 처리 방법에 대하여 제안하였다. 먼저 이벤트-프로파일 모델을 제시하고 효율적인 이벤트 검색을 위한 2단계 이벤트-프로파일 여과방법에 대하여 제안하였다.

향후 연구로는 정제단계에서 보다 효율적으로 검색을 수행할 수 있는 인덱스에 대한 연구가 필요하다. 또한 최종 구현을 수행하고 기존의 시스템과의 성능평가를 통해 제안하는 시스템의 우수성을 입증할 것이다.

참고문헌

- [1] S.Babum J,Widom. Contnuous Queries over Data Streams, SilGMOD Record, Sept. 2001
- [2] J.Chen, D.J.DeWitt, F.Tian, Y.Wang., NiagaraCQ: A Scalable Continuous Query System for Internet Databases, In Proc. SIGMOD, 2000
- [3] A.Hinze, A-MEDIAS: Concept and Design of an Adaptive Integrating Event Notification Service, PhD Thesis, Freie University Berlin, 2003, 12
- [4] 백승천. 트리거를 이용한 연속 질의 지원, 서울대학교 대학원 전기컴퓨터 공학부, 2004, 08
- [5] 이근주. 트리거와 저장프로시저를 이용한 연속질의 처리 시스템 설계 및 구현, 충남대학교 대학원 컴퓨터공학과 데이터&소프트웨어 공학, 2006, 02
- [6] 전성우, 김기학, 구훈영, 허홍식, 박종홍. RFID 기반 실시간우편 물류 기술, 전자통신동향분석, 제22권 3호, 2007, 06