

환자 모니터링 시스템을 위한 슬라이딩 윈도우 질의 모델 설계

김지수, 조대수

동서대학교

A Design of Sliding Window Query Model for Patient Monitoring System

Ji-Su Kim, Dae-Soo Cho

Dongseo University

E-mail : kimjs9307@nate.com, dscho@dongseo.ac.kr

요 약

기존의 데이터베이스 관리시스템(DBMS)은 스트림 데이터에 대한 연속질의를 고려하지 않고 설계되었기 때문에, 환자 모니터링 시스템과 같은 스트림 기반 응용에서의 요구를 만족하기 위해서는 새로운 질의 모델이 요구된다. 환자 모니터링 시스템은 혈압과 온도와 같은 다양한 유형의 생체 정보가 처리되어야 하며, 생체 센서로부터 측정된 생체 정보는 스트림으로 처리되어야 한다. 이 논문에서는 환자 모니터링 시스템에서 사용 될 수 있는 모든 형태의 질의를 4가지 유형으로 분류하였다. 그리고 이러한 4가지 유형의 질의를 표현 할 수 있는 새로운 슬라이딩 윈도우 질의 모델을 제안하였다.

ABSTRACT

A new query model is required to match requirements of stream-based applications such as patient monitoring system, since traditional DBMSs are not designed to provide continuous queries over stream data. In the patient monitoring system, there are many types of biomedical signals such as blood pressure and temperature, and these signals gathered by biomedical sensors should be treated as a stream, that is an ordered set of signals. In this paper, we categorized all possible queries to be used in patient monitoring system by four types of queries. Then, we have proposed a new sliding window query model which is capable of expressing these four types of queries.

키워드

데이터 스트림, 슬라이딩 윈도우, 연속질의

1. 서 론

전통적인 데이터베이스는 영속적인 데이터 저장과 복잡한 질의 처리를 요구하는 응용에서 사용되어 왔다. 그러나 최근 현재 데이터 모델에서는 적당하지 않고 특정영역에 대한 질의 처리에서도 적당하지 않은 응용들이 발생하고 있다. 예를 들어 센서 데이터, 인터넷 트래픽 정보, 주식 거래 정보, 웹 로그, 전화 통화 기록 등이 있다 [1]. 이러한 환경에서 정보는 연속적인 데이터의 형태로 발생한다.

데이터 스트림은 실시간으로 끊임없이 순서적으로(도착시간에 의해 암시적으로 또는 타임스탬프에 의한 명시적인 방법으로) 도달하는 데이터 아이템들의 연속이다. 데이터 아이템들이 도달하

는 순서를 제어하는 것이 불가능하며, 스트림의 전체를 저장하는 것은 비효율적이다. 또한 스트림에 대한 질의 역시 일정기간동안에 걸쳐 연속적으로 수행되며, 새로운 데이터가 도달할 때마다 새로운 결과 값을 제공하는 특징이 있다. 이 질의를 연속(continuous)질의라고도 한다[2]. 전통적인 데이터베이스 관리 시스템(DBMS)은 개별 데이터에 대해 신속하고 연속적으로 로딩하는 형태로 디자인 되어 있지 않으며, 데이터 스트림 응용에서 요구하는 연속질의에 대해서도 지원하지 않는다. 따라서 데이터 스트림 응용에서 DBMS를 이용하여 처리하는 것은 공간 효율 면이나 질의에 대한 응답 면에서도 매우 비효율적이다.

본 논문에서는 환자 모니터링 시스템에서의 연속질의를 다룬다. 환자 모니터링 시스템은 혈

압과 온도와 같은 환자의 다양한 생체 데이터를 센서로부터 실 시간적으로 감지하여 처리를 필요로 한다. 환자 모니터링 시스템에서 센서로부터 측정된 데이터는 스트림으로 처리되어야 하며 연속질의가 제공되어야 한다. 질의는 기존의 고정된 데이터 집합에 전달되는 질의와는 달리, 무한한 스트림을 대상으로 하기 때문에 질의의 대상을 제한하기 위해 윈도우와 수정 간격이 함께 시스템에 주어져야 하며, 연속질의의 생존기간을 명시 할 수 있어야 한다. 또한 사용자가 요구하는 사용자 정의 프레디카트(predicate)에 대해 지원 할 수 있어야 한다.

본 논문에서는 환자 모니터링 시스템에서 발생하는 데이터에 대한 연속질의 처리를 위해 시스템 환경을 분석하여 사용될 수 있는 모든 형태의 질의를 4가지 유형으로 분류하고 이를 처리하는 새로운 슬라이딩 윈도우 질의를 제안하고자 한다.

II. 관련 연구

데이터스트림 응용을 효과적으로 지원하기 위한 다양한 연구가 진행되어 왔다[3]. 현재 진행 중인 프로젝트들은 다음과 같다.

- 브라운대학과 MIT 대학 : Aurora(센서 모니터링, Data-flow)[4]
- 스탠포드 대학 : STREAM(범용 DSMS)[5]
- 버클리 대학 : Telegraph(가변적이고 예측하기 힘든 환경에서 질의를 효과적으로 처리하기 위한 적응적 질의 엔진)[6]
- OGI/Wisconsin 대학 : Niagara(인터넷 XML database)[7]
- Cornell 대학 : COUGAR(센서 database)[8]

현재 진행되고 있는 프로젝트에 사용되는 질의 언어들을 간단하게 정리하고, 연속 질의를 처리하기 위해 데이터 스트림에서 요구하는 윈도우 모델에 관하여 정리한다.

2.1 질의 언어

각 프로젝트에서 사용되는 질의 언어는 관계 기반, 객체 기반, 절차적 언어로 나누어 진다.

관계 기반 언어는 CQL, StreaQuel 등이 있다. CQL(Continuous Query Language)은 STREAM 프로젝트에서 사용되며, 슬라이딩 윈도우와 질의 결과를 스트림으로 변환하기 위한 변환 연산자를 제공한다. CQL은 PARTITION절을 이용하여 윈도우의 크기와 형태를 지정 할 수 있다. TelegraphCQ에서 사용되는 질의언어에서도 향상된 윈도우능력을 제공한다.

객체지향의 스트림 모델에 대한 하나의 접근은 스트림 구성요소를 타입계층에 따라 분류하

는 것이다. 이 방법은 Tribeca 프로젝트에서 사용되며, 인터넷 프로토콜 레이어를 계층적인 자료형으로 구현한다. 또 다른 모델은 COUGAR 프로젝트에서처럼 소스를 ADT로 모델화 하는 것이다. 질의 언어의 문법은 SQL과 유사하며 질의 재 수행 빈도를 나타내는 구문을 포함한다.

선언적인 질의 언어의 하나의 대안은 사용자가 데이터 흐름에 대하여 명시하도록 하는 것이다. Aurora 프로젝트의 절차적 언어에서, 사용자들은 GUI를 통해 상자들을 정렬하고, 자료의 흐름을 나타내는 화살표로 연결하여 질의 계획을 구성한다. Aurora는 다른 언어에서는 명시적으로 정의되지 않은 여러 연산자들을 포함하고 있다. 각 프로젝트에서 제안되고 있는 DSMS 질의 언어들의 요약은 표 1과 같다.

표 1. 질의 언어 요약

언어 / 프로젝트	허용 input	허용 output	새로운 연산자들	지원 윈도우	실행 빈도
CQL / STREAM	streams and relations	streams and relations	relation-to-stream, stream-to-relation	sliding	연속 or 주기
GSQL / Gigascope	streams	streams	order-preserving union	landmark	주기
SQuAl / Aurora	streams and relations	streams	resample, map, buffered sort	fixed, landmark, sliding	연속 or 주기
StreaQuel / TelegraphCQ	streams and relations	sequences of relations	Windows	fixed, landmark, sliding	연속 or 주기
Tribeca	single stream	streams	multiplex, demultiplex	fixed, landmark, sliding	연속

2.2 윈도우 모델 분류

많은 응용에서 데이터의 시간관계를 중요하게 여기며, 가장 최근에 데이터에 대해서 가장 흥미를 가지게 된다. 사용자가 가장 흥미를 가지는 데이터의 범위를 지정하기 위해 제안된 연산자가 윈도우 이다. 윈도우 범위에서 벗어난 데이터를 "제거된(expired)데이터"라 하며 윈도우를 처리할 때 더 이상 사용되지 않는다. 윈도우는 세 가지 기준에 따라 분류할 수 있다[9].

1. 양 끝점(endpoint)들의 이동 위치 : 두 개의 양끝 점을 고정하는 윈도우를 고정윈도우(fixed window), 양끝 점을 모두 이동하는 윈도우를 슬라이딩 윈도우(sliding window), 하나의 고정된 끝점과 하나의 이동하는 끝점을 경계윈도우(landmark window)로 정의 한다.
2. 물리적(Physical) vs 논리적(logical) : 물리적 윈도우는 시간에 의해 정의되며, 논리적 윈도우는 튜플의 개수에 의해 정의된다.
3. 수정 간격 : Eager reevaluation은 각각의 새로운 튜플이 도착할 때 마다 윈도우를 수정하지

만, 배치 처리(lazy reevaluation)는 “jumping window”을 도입한다. 만약 수정 간격이 윈도우의 크기보다 클 경우 그 결과는 서로 겹치지 않는 tumbling window의 연속이 된다[4].

III. 슬라이딩 윈도우 질의 모델

환자 모니터링 시스템에서는 환자의 현재 상태와 환자의 위급 상황을 판단하기 위한 특정 데이터 구분 및 사용자가 요구하는 사용자 정의 프레디카트 처리가 가장 중점이 된다. 센서는 심박, 혈압, 온도 등 여러 가지 유형의 데이터들로 측정될 수 있으며, 측정된 정보에는 센서를 고유하게 식별하는 ID도 포함된다. 또한 데이터 측정 빈도는 실시간, 주기적, 랜덤으로 측정될 수 있으나, 주기적(1초, 10초 등)으로 측정됨을 전제로 한다.

3.1 윈도우 모델

질의 대상을 제한하는 윈도우는 양 끝점이 모두 이동하는 슬라이딩 윈도우를 사용한다. 윈도우 형태는 시간 기반, 튜플 기반으로 설정 할 수 있으며, 크기는 사용자가 관심 있는 ‘최근 t시간 까지 도착한 데이터’ 또는 ‘최근에 도착한 n개의 데이터’등으로 정의한다. 윈도우 내에 존재하는 데이터들을 특정 단일 값으로 처리하는 집합연산을 사용한다. 예를 들어 집합연산으로는 MAX, MIN, AVG 연산 등이 있다. 집합 연산의 결과는 새로운 데이터가 들어올 때마다 결과 값을 수정하는 Eager reevaluation으로 수행한다.

3.2 질의 생존 기간과 실행 간격

환자 모니터링 시스템에서 연속질의를 정의할 때 질의 생존 기간과 실행 간격이 시스템에 제공된다. 생존 기간은 질의가 수행 하게 될 기간을 의미한다. 생존 기간은 사용자가 명시적(1분, 데이터 5개 등)으로 선언할 수 있으며, 실행 중인 질의는 사용자가 명시적으로 종료할 수도 있다. 실행 간격은 질의가 종료될 때까지 사용자에게 결과 값을 반환하는 간격을 의미한다. 실행 주기는 사용자가 명시적(5초마다)으로 선언할 수 있으며, 선언하지 않을 경우 새로운 데이터가 들어올 때마다 결과 값을 반환한다.

3.3 시간과 프레디카트

데이터 스트림에서 시간관계를 중요하게 여기며, 시스템에서 사용되는 모든 시간정의는 다음과 같다. 윈도우 크기는 시간으로 정의할 수 있으며, 정의된 윈도우 사이즈 범위는 $[t_{now}-S, t_{now}]$ 로 지정한다. t_{now} 는 현재시간, S는 윈도우 사이

즈를 의미한다. 질의 생존 기간은 $[t_q, t_q+I)$ 로 정의한다. t_q 는 질의 생성 시간, I는 사용자가 명시하는 질의 생존 기간을 의미한다.

프레디카트는 특정 조건을 만족하는 데이터들을 필터링 하기 위해 설정될 수 있다. 프레디카트의 범위 연산자는 $\langle, \leq, =, \rangle, \geq$ 등이 있으며, AND 와 OR를 사용할 수 있다.

3.4 질의 유형 분류

표 2. 질의 유형

질의 대상 필터링	단 일	다 중
무	첫 번째 유형(기본)	세 번째 유형(기본, F)
유	두 번째 유형(기본, P)	네 번째 유형(기본, P, F)

표 3. 각 유형별 API

첫 번째 유형
Query(NodeIDList, WindowValueType=TUPLE, WindowSize=1, QueryExecutionInterval, QueryLifeTime)
두 번째 유형
Query(NodeIDList, WindowValueType=TUPLE, WindowSize=1, ValueConditionType, FirstConditionValue, SecondConditionValue, QueryExecutionInterval, QueryLifeTime)
세 번째 유형
Query(NodeIDList, WindowValueType, WindowSize, AggregationFuctionType, QueryExecutionInterval, QueryLifeTime)
네 번째 유형
Query(NodeIDList, WindowValueType, WindowSize, AggregationFuctionType, ValueConditionType, FirstConditionValue, SecondConditionValue, QueryExecutionInterval, QueryLifeTime)

- NodeIDList : 센서 노드 리스트
- WindowValueType : 윈도우 형태
- WindowSize : 윈도우 크기
- AggregationFuctionType : 집합연산 형태
- ValueConditionType : 프레디카트 형태
- FirstConditionValue : 첫 번째 프레디카트 값
- SecondConditionValue : 두 번째 프레디카트 값
- QueryExecutionInterval : 질의 실행 주기
- QueryLifeTime : 질의 생존 기간

환자 모니터링 시스템에서 요구하는 모든 질의 형태를 질의 대상, 필터링 유무로 분류할 수 있다. 질의 대상은 윈도우 크기로 분류된다. 윈도우에 존재할 수 있는 최대 튜플의 개수를 기준으로 하여 최대 하나만 존재하는 것과 2개 이

상 존재하는 것으로 분류할 수 있다. 현재 상태에 관련된 연속질의 질의 생성 후 가장 최근에 측정된 하나의 데이터만 처리하면 된다. 시간 기반의 윈도우는 센서에서 측정되는 빈도에 따라 최대 튜플 개수가 가변적으로 변화하기 때문에 2개 이상 존재하는 것으로 분류한다. 크기가 1인 튜플 윈도우에 대해 집합연산을 처리할 수 있으나, 처리결과는 의미가 없으므로 지원하지 않는다. 필터링은 사용자 프레디키트의 설정 유무로 분류한다.

질의대상과 필터링 유무를 고려하여 분류할 경우 4가지 유형으로 나눌 수 있다. 질의 표현 형태는 모두 API로 나타내며, 4가지의 질의 유형은 표 2, 각 유형별의 API 형태는 표 3과 같다. 기본적으로 윈도우, 질의 실행 주기, 질의 생존기간은 모든 유형에 제공되며, F는 집합연산, P는 프레디키트를 의미한다.

모든 질의는 일시적인 질의 또는 연속 질의로 처리 될 수 있다. 그러나 첫 번째와 두 번째 유형은 일시적인 질의로, 세 번째와 네 번째 유형은 연속 질의로 처리될 빈도가 높다. 각 유형별로 사용될 수 있는 간단한 실 예제는 다음 표 4와 같다(간단하게 온도에 대한 질의 표현).

표 4. 각 유형별 간단한 실 예제

첫 번째 유형
· 1,3,5번 센서의 현재온도 출력
· 1번 센서의 현재온도를 10초 동안 출력
· 2,3번 센서의 현재온도를 5초단위로 1분 동안 출력
두 번째 유형
· 1,5번 센서의 현재온도가 37도보다 큰 값일 경우 출력
· 3,4,5번 센서의 현재온도가 37도보다 큰 값들을 1분 동안 출력
세 번째 유형
· 2번 센서의 5초 동안의 평균온도를 출력
· 1,2,5번 센서의 5개 데이터의 평균온도를 1분 동안 출력
· 3번 센서의 5초 동안의 평균온도를 10초단위로 1분 동안 출력
네 번째 유형
· 5번 센서의 5개 데이터의 평균온도가 37도보다 큰 값일 경우 출력
· 1,4번 센서의 5초 동안의 평균온도가 37도보다 큰 값들을 1분 동안 출력

IV. 결 론

본 논문에서는 현재 관심이 높아지고 있는 데이터 스트림과 데이터 스트림을 처리하기 위해 요구되어지는 연속질의와 윈도우에 대하여 살펴 보았다. 환자 모니터링 시스템에서 연속질의를 처리하기 위해 환경을 분석하고 4가지 유형의 슬라이딩 윈도우 질의를 제안하였다.

향후, 본 논문에서 고찰한 내용을 바탕으로 환자 모니터링 시스템을 구현하고 실험 평가를 해

야 할 필요가 있다. 시스템 구현 시 다중 센서에서 측정되는 데이터들을 저장하는 버퍼링 방법으로는 통합 버퍼, 질의당 버퍼를 할당하는 방법 등 여러 가지가 있을 수 있으며, 성능 향상을 고려해서 구현되어야 할 것이다. 또한 다중 연속질의를 보다 효율적으로 처리하기 위한 질의 최적화 방법에 대한 연구가 필요하다.

참고 문헌

- [1] Brian Babcock, Shivnath Babu, Mayur Datar, Rajeev MotWani, Jennifer Widom, "Models and Issues in Data Stream Systems"
- [2] JianJun Chen, David J. DeWitt, Feng Tian, Yuan Wang, "NiagaraCQ: a scalable continuous query system for Internet databases"
- [3] L.Goalb, M.Tamer Özsu, "Processing Sliding Window Multi-Join in Continuous Queries over Data Stream"
- [4] Don Carney, Ugur ? tintermel, Mitch Cherniack, Christian Convey, Sangdon Lee, Greg Seidman, Michael Stonebraker, Nesime Tatbul, Stan Zdonik, "Monitoring Streams - A New Class of Data Management Applications"
- [5] Arvind Arasu, Shivnath Babu, Jennifer Widom, "An Abstract Semantics and Concrete Language for Continuous Queries over Streams and Relations"
- [6] Sirish Chandrasekaran, Owen Cooper, Amol Deshpande, Michael J.Franklin, Joseph M.Hellerstein, Wei Hong, Sailesh Krishnamurthy, Sam Madden, Vijayshankar Raman, Fred Reiss, Mehul Shah, "TelegraphCQ: Continuous Dataflow Processing for an Uncertain World+"
- [7] Jianjun Chen, David J.Dewitt, Feng Tian, Yan Wang, "NiagaraCQ: A Scalable Continuous Query System for Internet Databases"
- [8] Philippe Bonnet, Johannes Gehrke, Praveen Seshadri, "Towards Sensor Database Systems"
- [9] Lukasz Golab and M.Tamer Özsu, "data stream management Issues - A Survey"