공간 데이터스트림을 위한 조인 전략 및 비용 모델

(Strategies and Cost Model for Spatial Data Stream Join)

유 기 현* 남 광 우**

(Ki Hyun Yoo) (Kwang Woo Nam)

요 약 GeoSensor 네트워크란 지리공간상에서 발생하는 다양한 현상들을 모니터링하는 특정형태의 센서네트워크 인프라 및 관련 소프트웨어를 의미한다. 그리고 이러한 GeoSensor 네트워크는 데이터스트림과 공간 속성의 데이터를 가진 스트림, 또는 공간 릴레이션과의 조합으로 구현될 수 있다. 하지만, 최근까지 연구된 센서 네트워크 시스템은 공간 정보를 배제한 센서 데이터스트림에 대한 저장 및 검색 방안 연구에 치중되어 있다. 따라서 본 논문은 GeoSensor 네트워크에서 데이터스트림과 공간 데이터가 결합된 형태의 공간 데이터스트림의 정의 및 그들 간의 조인 전략들을 제안한다. 본 논문에서 정의하고 있는 공간 데이터스트림에는 이동 객체 형태의 동적 공간 데이터스트림과 고정된 형태의 정적 공간 데이터스트림이 있다. 동적 공간 데이터스트림은 GPS와 같이 동적으로 이동하는 센서에 의해 전송되는 데이터스트림을 말한다. 반면, 정적 공간 데이터스트림은 일반 센서 형태의 데이터스트림과 이러한 센서들의 위치 값을 가지고 있는 릴레이션과의 조인으로 만들어 진다. 본 논문은 동적 공간 데이터스트림과 정적 공간 데이터스트림의 조인 및 조인 비용을 추정하는 모델을 제안하고 있다. 또한, 실험을 통해 제안하는 비용 모델의 검증 및 조인 전략에 따른 조인 성능을 보이고 있다.

키워드 : 지오센서 네트워크, 데이터스트림, 공간 조인

Abstract GeoSensor network means sensor network infra and related software of specific form monitoring a variety of circumstances over geospatial. And these GeoSensor network is implemented by mixing data stream with spatial attribute, spatial relation. But, until a recent date sensor network system has been concentrated on a store and search method of sensor data stream except for a spatial information. In this paper, we propose a definition of spatial data stream and its join strategy model at GeoSensor network, which combine data stream with spatial data. Spatial data streams defining in this paper are dynamic spatial data stream of a moving object type and static spatial data stream of a fixed type. Dynamic spatial data stream is data stream transmitted by moving sensor as GPS, while static spatial data stream is generated by joining a data stream of general sensor and a relation with location values of these sensors. This paper propose joins of dynamic spatial data stream and static spatial data stream, and cost models estimating join cost. Finally, we show verification of proposed cost models and performance by join strategy.

Key words : GeoSensor Network, Data Stream, Spatial Joins

1. 서론

GeoSensor 네트워크란 지리공간상에서 발생하는 다양한 현상들을 모니터링하는 특정형태의 센서네트워크 인프라 및 관련 소프트웨어를 의미한다. 여기에는 데이터를 수집하고, 집계하며, 분석하며, 공간적 시간적 상황에 따라 적절한 반응을 주기 위한 다양한 연구들이 포함될 수 있다.

GeoSensor 네트워크의 주요 응용들의 예를 들자면, ① 환경 모니터링(동식물 생태, 건물, 홍수 탐지, 토양 및 습기, 대기/해양 모니터링), ②객체 추적(차량/동물 추적, 군사, 물류), ③객체 감시(응급 의료, 침입 탐지, 지진위험, 산림 방재) 분야 등이 있다. 이들 센서 네트워크 응용의 공통점은 사용자 및 애플리케이션 레벨에서 실시간으로 데이터를 모니터링하거나 분석하기 위해 필요한 공간에 센서 장비를 설치하고, 주기적인 시간마다 데이터를 수집하는 방식을 취한다는 것이다. 따라서 센서 스트림 데이

† 이 논문은 2006년 정부(교육인적자원부)의 재원으로 한국학술진흥재단의 지원을 받아 수행된 연구임"(KRF-2006-311-D00770)

* 군산대학교 컴퓨터정보공학과 석사과정, ykhid@kunsan.ac.kr

** 군산대학교 컴퓨터정보공학과 조교수, kwnam@kunsan.ac.kr(교신저자)

논문접수 : 2008.08.18 수정일 : 1차 2008.12.08 심사완료 : 2008.12.11

터는 인접한 시간과 공간 사이에 강한 시공간적 의존성과 인과관계 특성을 가지며, 센서 네트워크 미들웨어에 수집되는 센서 스트림 데이터는 일련의 센서 데이터 스트림들에 대해 시공간 통합적인 측면에서 종합적으로 분석되어야 한다. 이것은 기존의 센서 네트워크 시스템이 단순한 데이터 스트림의 정제, 축소, 집계, 요약 등과 같은 단순 데이터 계산형 시스템에서 필연적으로 시·공간적 의미들을 포함한 시스템의 형태로 확장될 필요성이 있음을 의미한다.

최근까지 센서 네트워크에서 발생하는 데이터 스트림 처리를 위한 다양한 연구들이 있었다. 대표적인 연구들은 센서 네트워크내에서 질의를 처리하는 In-Network 질의 처리 시스템인 COUGAR[1], 데이터 스트림 서버 상에 질의를 처리하는 데이터 스트림 관리 시스템들인 관계형 DB 모델 기반의 STREAM[2]과 객체 DB 모델 기반의 Aurora[3] 및 Tribeca[4], 불확실성을 지원하는 TelegraphCQ [5] 등이 있다. 그러나 이러한 시스템들은 공간 데이터 스트림 질의 처리를 지하지 않으며, GeoSensor 스트림 처리에 대한 필요성 제기[6]에도 불구하고 아직까지 데이터 스트림 관리 시스템에서 공간센서 스트림에 대한 처리에 대한 연구는 극히 제한적으로 수행되고 있다.

본 논문은 GeoSensor 네트워크 및 스트림 데이터베이스 연구에서 아직 제기되지 않았으며 해결되어야하는 다양한 문제들중에서 공간적 정보를 가진 공간 데이터스트림의 정의 및 공간 데이터스트림간 조인 전략들과 그에 따른 조인 전략들 간의 비용을 추정하는 비용 모델을 제시한다.

본 논문의 구성은 다음과 같다. 2장에서는 기존의 각 스트림과 공간 데이터에 대한 조인 기법들, 그리고 관련 연구들을 살펴보고, 3장에서는 공간 데이터스트림 정의 및 조인 기법들 간의 조인 전략들을 제안한다. 4장에서는 각 조인 전략들의 구성 요소로 단방향 조인 알고리즘들에 대한 비용 모델을 제시한다. 5장에서는 다양한 실험 및 성능 평가를 보이고, 마지막 6장에서 결론 및 추후 연구해야 할 방향에 대해 언급한다.

2. 관련 연구

2.1 DSMS(Data Stream Management System)

센서들로부터 수집되는 센서 데이터는 매우 빠른 속도로 추가되고 한번 추가된 데이터는 삭제되거나 수정되지 않는 특징이 있다. 이러한 특징을 갖는 데이터를 실시간으로 처리하기 위해 데이터 스트림(Data Stream)이라는 새로운 형태의 데이터 모델에 대한 연속 질의(Continuous Query)를 처리할 수 있는 DSMS가 필요하게 되었다[7]. 이러한 DSMS는 디스크나 메모리에 저장되어 있는 모든 데이터를 질의 처리 대상으로 하는 기존 DBMS와는 다르게, 최근 데이터만을 처리하거나 과거부터 현재까지 입력된 데이터에 대한 요약 정보만을 처리한다. 즉, 디스크 등의 저장소에 쌓여있는 데이터를 메모리에 로드하는 대신 실시간으로 입력되는 데이터를 처리 대상으로 하며, 질의는 한 번만 수행되는 것이 아니라 시스템에 등록된 후 연속적으로 수행된다. 이러한 DSMS는 센서 네트워크뿐만 아니라 네트워크 모니터링, 결제 정보 분석, 판매 내역 트랜잭션 분석 등과 같은 다양한 분야에서 활용이 가능하다.

2.2 스트림 조인

스트림 데이터를 조인하기 위해서는 스트림 데이터의 특성을 고려하여 요구되는 것들이 있다. 예를 들면, 연속적인 스트림 데이터의 생성 때문에 발생하는 메모리 요구, 스트림 데이터 환경의 non-blocking 연산자, Sliding Windows, 혹은 스트림 데이터를 처리하기 위한 Batch Processing, Sampling, Synopses 등의 기법들이다. 이러한 기법들 중에서 Sliding Windows라는 하나의 윈도우를 적용하여 스트림 데이터를 조인하는 기법들이 많이 연구되고 있다. 다수의 스트림 데이터를 조인하기 위해 Window 조인 방법이 사용되었다[8,9,10]. Telegraph project는 multi-way 조인 연산자 구현을 제안했다[10]. 이 방법에서, 윈도우가 튜플 수의 관점에서 정의되고 각 스트림에서 새로운 튜플이 오래된 퓨틀을 제거하도록 강요된다. [8]는 시간 윈도우 개념을 사용한다. 또한, 두 스트림의 도착율이 다른 스트림들의 윈도우 조인을 다루는 방법도 연구되었다[9]. [11]은 실행 비용을 줄이기 위해 비대칭 조인 방법을 사용한다. 예를 들면, 하나의 스트림에 대해서 세미 중첩 루프 조인을 사용하고, 다른 스트림에서는 세미 해시 조인을 사용하여 그것들의 조합으로 전체 조인 비용을 줄였다. Batch Processing 기법을 사용한 XJoin은 Symmetric Hash Join을 한다[12]. [13]은 연속질의를 효율적으로 처리하기위해 공간조인 기반 연속질의 처리 알고리즘을 제안했다. 이 방법은 다차원 공간상에서 연속질의를 처리한다.

2.3 공간 조인

두 개의 공간 데이터들에 대해 인덱스가 존재 하는 경우 보통 이러한 인덱스들을 사용하는 것이 더욱 효과적이다. 그러나 때때로 양립할 수 없는 타입의 인덱스들이 존재 할 수 있다. Corral 등은 이러한 경우 하나의 인덱스를 무시하고 인덱스 중첩 루프 조인을 수행하는 것을 제안했다[14].

하나의 공간 데이터에만 인덱스가 있을 경우, 인덱스 중첩 루프 조인을 사용하여 공간 조인이 수행 될 수 있다. 다른 방법으로 인덱스가 없는 공간 데이터에 대해 인덱스를 구성하여 두 개의 인덱스들을 조인하는 방법이 있다. 만약 하나의 공간 데이터에만 인덱스가 있다면, 다른 공간 데이터는 부피-적재 기술(bulk-loading technique)을 사용하여 효율적으로 인덱스 될 수 있다[15]. 이 방법은 그 인덱스가 저장되고 나중에 재사용될 때 유용하다. 이렇게 구성되는 인덱스는 본래 가지고 있던 인덱스의 구조와 흡사하도록 만들어진다. Lo and Ravishankar는 Seeded-Tree를 제안했다[16]. 이 방법은 인덱스를 구성하기 위하여, Seed Level이라 부르는 본래

가지고 있던 인덱스의 상위 레벨을 사용한다. 이 상위 레벨을 사용하여 인덱스가 없는 공간 데이터를 분할한다. 그 데이터가 분할되면, 부피-적재 기술(bulk-loading technique)을 사용하여 하나의 R-Tree로 변환된다.

3. 공간 데이터스트림 조인 모델

3.1 GeoSensor 네트워크에서 공간 조인

GeoSensor 데이터스트림 시스템에서 S를 스트림이라 하고 R을 릴레이션이라고 할 때, 공간 조인의 대상이 될 수 있는 요소들은 다음과 같이 정의할 수 있다.

S_{data} : 일반적인 데이터스트림으로서, 공간 정보를 포함하지 않은 데이터스트림이다.

$S_{static_{geo}}$: 정적 공간 데이터스트림으로서, 공간 정보를 포함하지 않은 일반적인 형태의 센서 데이터스트림과 이러한 센서들의 위치 정보가 저장되어 있는 릴레이션과의 조인으로 만들어진 공간 데이터스트림이다.

$S_{moving_{geo}}$: 이동체 데이터스트림으로서, GPS 위치와 같이 동적으로 이동하는 센서에 의해 전송되는 공간 데이터스트림이다.

R_{geo} : 공간 릴레이션으로서, 데이터베이스 내에 지리정보를 저장하고 있는 릴레이션이다.

$R_{static_{geo}}$: 단순히 공간 좌표를 가지고 있는 릴레이션으로서, $S_{static_{geo}}$에서 공간 정보를 포함하지 않은 센서 데이터스트림에 공간 정보를 포함시키기 위해 필요한 릴레이션이다.

GeoSensor 릴레이션과 스트림이 위와 같다고 할 때 GeoSensor 데이터스트림 시스템에서 처리해야하는 조인의 종류와 예는 표 1과 같다. 표 1을 보면 조인의 종류를 크게 공간 데이터스트림간 조인과 공간 데이터스트림과 공간 릴레이션간 조인 두 가지로 나누었고, 이것들을 다시 좀 더 구체적인 5가지의 조인으로 분류하였다. 공간 데이터스트림간 조인은 공간 데이터스트림들인 $S_{static_{geo}}$와 $S_{moving_{geo}}$로만 이루어진 조인이다. 반면, 공간 데이터스트림과 공간 릴레이션의 조인은 공간 데이터스트림들인 $S_{static_{geo}}$와 $S_{moving_{geo}}$, 공간 릴레이션인 R_{geo}로 이루어진 조인들이다. 본 논문은 표 1에서 공간 데이터스트림간 조인에 대한 비용 모델 및 조인 전략들을 제시하고 실험을 통하여 조인 전략에 따른 조인 성능을 측정한다.

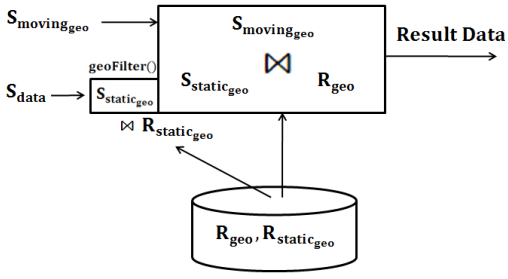


그림 1. GeoSensor 네트워크 공간 조인 프레임워크

공간 데이터스트림 질의는 그림 1과 같이 $S_{static_{geo}}$와 $S_{moving_{geo}}$, 그리고 R_{geo}의 데이터들에 대하여 수행된다. 이 때, $S_{static_{geo}}$는 공간 조인에서 수행되기 위해 실제 공간 데이터스트림으로 변환되는 geoFilter() 연산이 필요하다.

geoFilter() 연산은 정적 공간 데이터스트림인 $S_{static_{geo}}$를 만드는 연산이다. 공간 정보를 포함하지 않은 일반 센서 데이터스트림 S_{data}와 이러한 센서들의 위치 정보를 저장하고 있는 릴레이션인 $R_{static_{geo}}$와의 조인을 통해 공간 정보를 가지는 정적 공간 데이터스트림이 만들어 진다. 이 연산은 $R_{static_{geo}}$에 대해서 인덱스를 구성한 다음, 삽입되는 스트림데이터와의 조인 과정을 거친다.

3.2 공간 데이터스트림간 조인 비용 프레임워크

일반적인 데이터스트림 간의 조인 전략은 슬라이딩 윈도우 조인 방법을 많이 사용한다. 그림 2와 같이 데이터스트림 A와 B로부터 스트림데이터가 들어온다고 가정했을때, 데이터스트림 B로부터 새로운 데이터 엘리먼트 a가 입력되었을 때의 조인 수행 과정은 일반적으로, (1) 새로 들어온 데이터 엘리먼트와 데이터스트림 A의 슬라이딩 윈도우의 데이터 엘리먼트들 간에 조인 조건을 만족하는지 검사하는 과정(probe), (2) 새로운 데이터 엘리먼트를 데이터스트림 B에 삽입하는 과정(insert), (3) 데이터 엘리먼트의 삽입과 함께 데이터스트림 B의 슬라이딩 윈도우의 범위를 벗어난 데이터를 윈도우로부터 삭제

표 1. GeoSensor 데이터스트림 조인의 종류와 질의 예

GeoSensor 스트림 조인		질의의 예
공간 데이터 스트림간 조인	$S_{static_{geo}} \bowtie S_{static_{geo}}$	고정 온도 센서들 간의 거리가 1km 이내이면서 10도 이상 온도차가 나는 센서들을 보이시오
	$S_{static_{geo}} \bowtie S_{moving_{geo}}$	온도가 30도 이상인 센서로부터 1km 이내 지역을 지나가는 GPS 차량들의 30초 평균 속도를 보이시오
	$S_{moving_{geo}} \bowtie S_{moving_{geo}}$	120km이상의 속도로 500m이내의 거리를 두고 지나가는 GPS 차량들을 보이시오
공간 데이트 스트림과 공간 릴레이션간 조인	$S_{static_{geo}} \bowtie R_{geo}$	공원 1km이내의 고정 온도센서들의 최근 30초간 온도 평균을 보이시오
	$S_{moving_{geo}} \bowtie R_{geo}$	공원을 지나가는 GPS 차량들의 최근 30초간 평균 속도를 보이시오

하는 과정(invalidate)으로 이루어진다[13]. 본 논문은 이 슬라이딩 윈도우의 구조를 Grid와 R-tree 기반의 인덱스로 구성하여 조인을 수행한다. 이는 공간 좌표를 가진 공간 데이터스트림간의 조인을 좀 더 효율적으로 하기 위한 것이다.

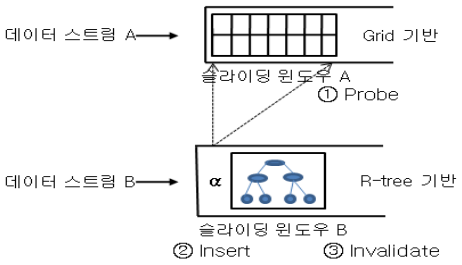


그림 2. 슬라이딩 윈도우 조인 전략

데이터스트림 A와 데이터스트림 B에 대한 슬라이딩 윈도우 조인 비용 공식은 아래와 같다[9].

$$C_{A\bowtie B} = \lambda a(probe(b) + insert(a) + invalidate(a)) + \lambda b(probe(a) + insert(b) + invalidate(b))$$

위 공식의 첫 번째 요소가 스트림 A에 대한 처리 비용을 측정한다. 그리고 단위 시간당 도착 튜플들을 각 요소에 곱해준다. (ex. λ_a, λ_b) 위의 공식에서 조인 연산의 비용을 두 개의 독립적인 서브그룹으로 나눌 수 있다.

$$C_{A\bowtie B} = C_{A\times B} + C_{A\times B}$$
$$C_{A\times B} = \lambda a(probe(b)) + \lambda b(insert(b) + invalidate(b))$$
$$C_{A\times B} = \lambda b(probe(a)) + \lambda a(insert(a) + invalidate(a))$$

본 논문에서 제안하는 조인 비용 모델은 위 두 개의 세미 조인에 각각 적용되어 그들의 합으로 전체 조인 비용을 구한다.

본 논문은 표 1에서 정의한 조인들 중 공간 데이터스트림간 조인을 다루고 있다. 때문에 표 1에서의 공간 데이터스트림과 공간 릴레이션간 조인은 정의만 하였을 뿐 본 논문에서는 다루지 않고 있다.

본 논문에서 다루고 있는 공간 데이터스트림간 조인에서 $S_{static_{geo}} \bowtie S_{static_{geo}}$ 조인을 분배법칙을 적용하여 최적화 할 수 있다.

$$S_{static_{geo}} \bowtie S_{static_{geo}} = (S_{data_1} \bowtie R_{static_{geo}}) \bowtie (S_{data_2} \bowtie R_{static_{geo}}) = (S_{data_1} \bowtie S_{data_2}) \bowtie R_{static_{geo}}$$

이렇게 분배법칙을 적용하면, 2번의 조인 수행으로 비용을 줄일 수 있는 이점이 있다.

3.3 공간 데이터스트림간 조인 전략

본 논문에서는 공간 데이터스트림간 조인 전략을 위해서 Nested Loop 조인(NLJ), Grid 조인(GJ), R-tree 조인(RJ) 방법을 이용한다. 각 조인 방법들은 3.2절에서 언급했던 세미 조인으로 분할되고, 이 세미 조인들의 조합으로 공간 데이터스트림간 조인 전략이 만들어 진다. 4장에서 분할된 세미 조인의 비용을 추정하는 비용 모델을 제안하고 있는데, 이 분할된 세미 조인을 단방향 조인이라고 명시 했다. 조인 전략의 예를 들면, 다음과 같다.

$$C_{A\bowtie B} = C_{A\times B} + C_{A\times B}$$
$$C_{A\bowtie B} = C^S_{A\times B}(NLJ) + C^S_{A\times B}(GJ)$$

이러한 방법으로 9가지의 조인 전략들이 만들어 질 수 있다. 5장에서는 실험을 통하여 이러한 조인 전략들의 비용을 비교, 분석 한다.

4. 공간 데이터 스트림 조인의 단위 비용

이 장에서는 3장에서 다루는 조인 전략들의 비용을 추정하기 위한 비용 모델을 제안한다. 표 2는 이 논문에서 사용되고 있는 비용 모델을 위한 심벌들을 정리한 것이다.

표 2. 비용 모델을 위한 심벌과 정의

심벌	정 의
dim	차원 수(2차원 기준)
D	데이터 집합의 밀도
D_k	차원 k에서의 데이터 집합의 밀도
N	그리드 해상도
M	데이터 집합의 객체 수
f	평균 R-tree 노드 팬아웃
q_k	차원 k에서의 질의 사각형 q의 평균 범위
$D_{B,l}$	레벨 l에서 R-tree B_i의 데이터 집합의 밀도
λ_b	스트림 B의 도착률
B	슬라이딩 윈도우 B에서의 튜플 수
P_d	조인 알고리즘 d의 검색 가중치 요소
I_d	조인 알고리즘 d의 갱신 가중치 요소

4.1 단방향 Nested Loop 조인(NLJ)의 비용

단방향 Nested Loop 조인은 일반 Nested Loop 조인의 세미 조인을 뜻한다. Nested Loop 조인을 사용하는 이유는 이 Nested Loop 조인이 조인 연산을 수행하는데 있어서 가장 기본적인 조인 방법이기 때문이다. 따라서 가장 많은 조인 비용이 든다. 본 논문에서는 다른 조인들과의 비용을 비교하기 위해서 Nested Loop 조인을 사용한다. A의 B에 대한 단방향 NLJ의 비용은 다음과 같다[9].

$$C^S_{A\times B}(NLJ) = \lambda_a B \times P_N + 2\lambda_b \times I_N$$

where P_N= NLJ 검색 가중치
I_N= NLJ 갱신 가중치

위 공식은 3장에서 다룬 세미 조인 비용 공식을 NLJ 알고리즘에 적용하여 만들어진 것이다. 세미 조인 비용은 슬라이딩 윈도우의 검색 비용과, 삽입, 삭제 비용이 더해져 구해진다. $\lambda_a B$는 단방향 NLJ의 검색 비용을 나타내

고, λ_b는 단방향 NLJ의 삽입 비용을 나타낸다. 본 논문에서의 슬라이딩 윈도우 조인은 튜플 수에 기반하기 때문에 윈도우에 삽입과 동시에 삽입한 만큼 윈도우에서 삭제가 일어난다. 따라서 $2\lambda_b$가 삽입, 삭제 비용을 나타낸다. P_N과 I_N은 NLJ의 검색, 갱신 가중치로서 시스템 환경에 따라 다르게 나타날 수 있다.

4.2 단방향 Grid 조인(GJ)의 비용

단방향 Grid 조인 또한 일반 Grid 조인의 세미 조인을 뜻한다. 본 논문은 공간 속성을 가진 데이터스트림간의 조인을 위하여 공간 조인 기법 중 하나인 Grid 조인 기법을 세미 조인 형태로 변형한 단방향 Grid 조인 비용 모델을 제안한다. A의 B에 대한 단방향 GJ의 비용은 다음과 같다.

$$C^S_{A\times B}(GJ)$$
$$= \lambda_a \times \left(M_B \prod_{k=1}^{\dim} \left(\left(\frac{D_{Bk}}{B} \right)^{\frac{1}{\dim}} + N^{-1} \right) N^{\dim} \right) \times P_G$$
$$+ 2 \times \lambda_b \times \left(M_B \prod_{k=1}^{\dim} \left(\left(\frac{D_{Bk}}{B} \right)^{\frac{1}{\dim}} + N^{-1} \right) N^{\dim} \right) \times I_G$$

where P_G = GJ 검색 가중치위 공식은
I_G = GJ 갱신 가중치

위 공식은 4.1절과 마찬가지로 GJ 알고리즘을 세미 조인 형태로 변형한 것이다. 단방향 GJ 비용 또한 검색 비용과 삽입, 삭제 비용이 더해져서 구해진다. 위 식에서 $\left(M_B \prod_{k=1}^{\dim} \left(\left(\frac{D_{Bk}}{B} \right)^{\frac{1}{\dim}} + N^{-1} \right) N^{\dim} \right)$이 그리드의 검색 연산을 나타낸다[17]. 위 공식에서도 Grid의 검색과 갱신에 따른 가중치가 적용 된다.

4.3 단방향 R-tree 조인(RJ)의 비용

R-tree는 공간 데이터를 색인하는데 효율적인 구조를 가진다. 때문에 본 논문에서는 공간 데이터스트림간 조인을 위하여 단방향 R-tree 조인을 제안한다. A의 B에 대한 단방향 RJ의 비용은 다음과 같다.

$$C^S_{A\times B}(RJ)$$
$$= \lambda_a \times \sum_{l=1}^{1+\lceil \log_f \frac{B}{f} \rceil} \left\{ \frac{B}{f^l} \cdot \prod_{k=1}^{\dim} \left(\left(D_{B_i,l} \cdot \frac{f^l}{B} \right)^{\frac{1}{\dim}} + q_k \right) \right\} \times P_R$$
$$+ 2 \times \lambda_b \times \sum_{l=1}^{1+\lceil \log_f \frac{B}{f} \rceil} \left\{ \frac{B}{f^l} \cdot \prod_{k=1}^{\dim} \left(\left(D_{B_i,l} \cdot \frac{f^l}{B} \right)^{\frac{1}{\dim}} + q_k \right) \right\} \times I_R$$

where P_R = RJ 검색 가중치
I_R = RJ 갱신 가중치

위 식에서 $\sum_{l=1}^{1+\lceil \log_f \frac{B}{f} \rceil} \left\{ \frac{B}{f^l} \cdot \prod_{k=1}^{\dim} \left(\left(D_{B_i,l} \cdot \frac{f^l}{B} \right)^{\frac{1}{\dim}} + q_k \right) \right\}$는 R-tree의 검색 연산을 나타낸다[18]. 마찬가지로 R-tree의 검색과 갱신 가중치가 적용 된다.

4.5 가중치 요소 평가

초당 100개의 입력 튜플 수를 가지는 검색 작업에 대한 CPU 시간을 측정하기 위해, 우리는 일괄적으로 6000개의 튜플을 처리하고 전체 실행 시간을 측정했다. 측정된 가중치 요소를 가진 비용 공식은 아래와 같다.

$$C^S_{A\times B}(NLJ) = \lambda aB \times 3.5 \times 10^{-4} + 2\lambda b \times 10^{-4}$$

$$C^S_{A\times B}(GJ)$$
$$= \lambda_a \times \left(M_{Bi} \prod_{k=1}^{\dim} \left(\left(\frac{D_{Bk}}{B} \right)^{\frac{1}{\dim}} + N^{-1} \right) N^{\dim} \right) \times 6.2 \times 10^{-4}$$
$$+ 2 \times \lambda_b \times \left(M_{Bi} \prod_{k=1}^{\dim} \left(\left(\frac{D_{Bk}}{B} \right)^{\frac{1}{\dim}} + N^{-1} \right) N^{\dim} \right) \times 6.5 \times 10^{-4}$$

$$C^S_{A\times B}(RJ)$$
$$= \lambda_a \times \sum_{l=1}^{1+\lceil \log_f \frac{B}{f} \rceil} \left\{ \frac{B}{f^l} \cdot \prod_{k=1}^{\dim} \left(\left(D_{B_i,l} \cdot \frac{f^l}{B} \right)^{\frac{1}{\dim}} + q_k \right) \right\} \times 4.7 \times 10^{-4}$$
$$+ 2 \times \lambda_b \times \sum_{l=1}^{1+\lceil \log_f \frac{B}{f} \rceil} \left\{ \frac{B}{f^l} \cdot \prod_{k=1}^{\dim} \left(\left(D_{B_i,l} \cdot \frac{f^l}{B} \right)^{\frac{1}{\dim}} + q_k \right) \right\} \times 4.7 \times 10^{-4}$$

5. 공간 조인의 비용 실험 및 평가

5.1 실험 데이터 및 실험 환경

본 논문에서 사용하는 실험 데이터는 $S_{moving_{geo}}$와 $S_{static_{geo}}$이다. 이전 장에서도 언급했듯이 본 논문에서는 공간 데이터스트림간 조인 만을 다루기 때문에 두 개의 스트림 데이터들을 사용한다. 3장에서 정의한 것처럼, $S_{moving_{geo}}$는 GPS 위치와 같이 동적으로 이동하는 센서에 의해 전송되는 공간 데이터스트림이고, $S_{static_{geo}}$는 고정된 위치에 있는 공간 데이터스트림이다. $S_{static_{geo}}$는 S_{data}와 $R_{static_{geo}}$의 조인 연산인 geoFilter() 연산을 통해 만들어진다.

실험은 AMD Athlon 64X2 Dual 2.02GHz의 CPU와 1GB의 메모리, 그리고 윈도우 XP에서 수행했다.

5.2 실험평가

데이터스트림 시스템 하에서 데이터스트림들의 전송율은 상황에 따라 달라질 수 있다. 때문에 본 논문에서 제시하는 조인 전략들의 비용을 평가할 때 데이터스트림 A, B로부터 입력되는 스트림들의 비율을 고려할 필요가 있다. 따라서 입력되는 스트림들의 비율에 따른 비용을 실험을 통해 알아보았다. 또한 슬라이딩 윈도우의 크기에 따른 비용 변화 및 Grid 조인의 해상도에 따른 비용 변화에 대해서도 비교 평가하였다.

5.2.1 조인 전략들의 비용 비교

그림 3은 본 논문에서 제시하는 조인 전략들의 비용을 추정하는 그래프를 나타내고 있다. 스트림들의 비율은 데이터스트림 A, B에 대하여 λ_a/λ_b로 표현한다. 또한 데이터스트림 A, B의 슬라이딩 윈도우는 튜플기반의 윈도우로 5000의 크기로 측정했다. 9가지의 조인 전략 중 NLJ-

NLJ 조인 전략은 비용이 너무 커서 그래프에서 제외했다. 그림 3을 보면, Rtree-Rtree 전략과 Grid- Grid 전략은 스트림들의 비율에 상관없이 일관된 비용을 나타낸다. 이것은 실험에서 정의한 스트림들의 비율 때문이다. 여기서는 Rtree-Rtree 전략이 Grid-Grid 전략보다 비용이 적게 나타났지만, Grid의 해상도를 고려한 실험에서는 Grid 해상도에 따라 Grid-Grid 전략이 더 적은 비용을 나타냈다. 나머지 서로 다른 조인 조합들은 각각 서로 대칭의 결과를 보이고 있다.

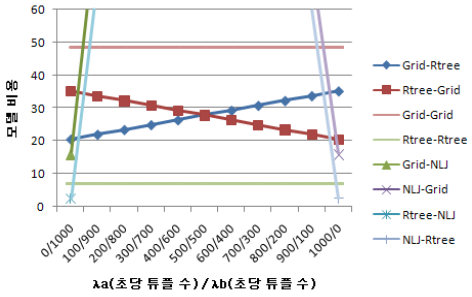


그림 3. 단방향 조인 조합 비용
(슬라이딩 윈도우 A, B = 5000)

5.2.2 슬라이딩 윈도우 크기에 따른 비용

이전 실험에서는 일정한 크기의 슬라이딩 윈도우로 비용을 측정하였다. 하지만 슬라이딩 윈도우의 크기 변화에 따른 비용변화를 고려할 필요가 있다. 이 실험에서는 각 스트림들의 비율은 같고, 윈도우의 크기만 달리하여 비용을 측정하였다.

그림 4는 슬라이딩 윈도우 크기 변화에 따른 비용을 측정한 그래프이다. 그림에서 보는 것처럼 Rtree-Rtree 전략은 윈도우의 크기에 영향을 적게 받는 반면, Grid-Grid 전략의 경우 윈도우 크기에 따라 비용 차이가 나는 것을 볼 수 있다. 또한 Rtree-Grid와 Grid- Rtree 전략들의 비용이 똑같은 것을 볼 수 있는데 이것은 스트림들의 비율이 같기 때문이다.

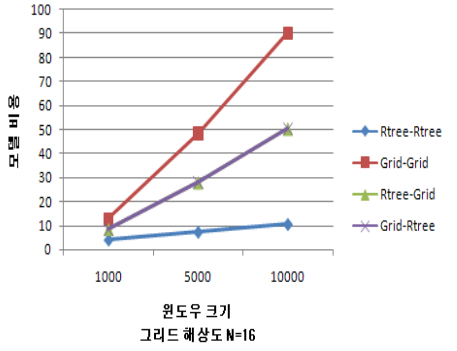


그림 4. 윈도우 크기에 따른 비용(N=16)

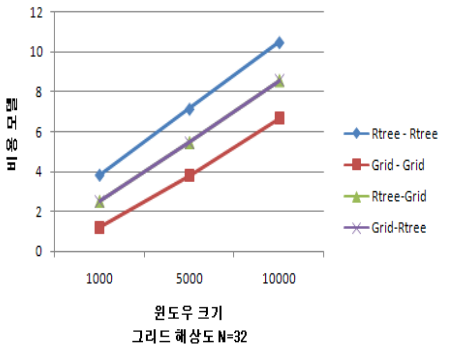


그림 5. 윈도우 크기에 따른 비용(N=32)

단락 5.2.1과 5.2.2를 보면 Rtree-Rtree 전략이 가장 적은 비용을 나타냈다. 하지만 Grid 조인의 경우 Grid 해상도에 따라 성능이 달라지기 때문에 Grid 해상도에 따른 비용을 측정해야 했다. 그림 5를 보면 Grid 해상도가 32 이상부터 Grid 조인의 비용이 현저하게 줄어드는 것을 알 수 있다.

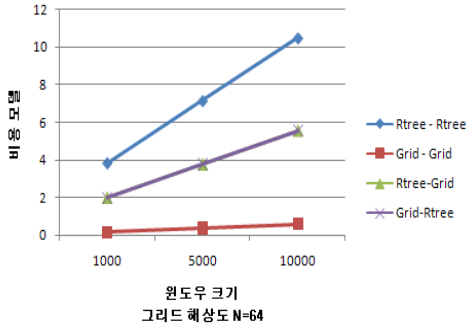


그림 6 윈도우 크기에 따른 비용(N=64)

이 단락에서 우리는 본 논문이 제시하는 조인 전략들의 몇 가지 비용들을 보았다. 실험을 통해 그리드 해상도가 일정한 값 이상 커지면 Grid-Grid 조인의 비용이 가장 적게 든다는 것을 알 수 있었다.

6. 결론

GeoSensor 네트워크란 지리공간상에서 발생하는 다양한 현상들을 모니터링하는 특정형태의 센서네트워크 인프라 및 관련 소프트웨어를 의미한다. 그리고 이러한 GeoSensor 네트워크는 데이터스트림과 공간 속성의 데이터를 가진 스트림, 또는 공간 릴레이션과의 조합으로 구현될 수 있다. 하지만, 최근까지 연구된 센서 네트워크 시스템은 공간 정보를 배제한 센서 데이터스트림에 대한 저장 및 검색 방안 연구에 치중되어 있다. 따라서 본 논문은 GeoSensor 네트워크에서 데이터스트림과 공간 데

이터가 결합된 형태의 공간 데이터스트림의 정의 및 그들 간의 조인 전략들을 제안한다. 또한 조인 전략에 대한 비용 모델을 제시한다. 실험을 통해 제시하는 비용 모델의 성능을 측정하였다.

데이터스트림 시스템 하에서 데이터스트림들의 전송율은 상황에 따라 달라질 수 있다. 때문에 본 논문에서 제시하는 조인 전략들의 비용을 평가할 때 데이터스트림 A, B로부터 입력되는 스트림들의 비율을 고려할 필요가 있다. 따라서 입력되는 스트림들의 비율에 따른 비용을 실험을 통해 알아보았다. 또한 슬라이딩 윈도우의 크기에 따른 비용 변화 및 Grid 조인의 해상도에 따른 비용 변화에 대해서도 비교 평가하였다.

본 논문의 향후 연구는 본 논문에서 정의만 하고 다루지 못했던 공간 데이터스트림과 공간 릴레이션과의 조인 전략이다. 향후 이 연구가 좀 더 진척이 된다면 공간 정보의 다양한 응용이 가능하리라 본다.

참 고 문 헌

[1] Y. Yao, J. Gehrke, "The Cougar Approach to In-Network Query Processing in Sensor Networks," ACM SIGMOD Record, Vol.31 No.3, 2002, pp. 9-1.

[2] A. Arasu, B. Babcock, S. Babu, M. Datar, K. Ito, I. Nishizawa, J. Rosenstein, J. Widom, "STREAM: The Stanford Stream Data Manager," IEEE Data Engineering Bulletin, Vol.26 No.1, 2003, pp. 19-26.

[3] D. Abadi, D. Carney, U. Cetintemel, M. Cherniack, C. Convery, S. Lee, M. Stonebraker, N. Tatbul, S. Zdonik, "Aurora: A New Model and Architecture for Data Stream Management," VLDB Journal, 2003.

[4] M. Sullivan, A. Heybey, "Tribeca: A System for Managing Large Databases of Network Traffic," In Proc. of USENIX Annual Technical, 1998.

[5] S. Chandrasekaran, O. Cooper, A. Deshpande, M. J. Franklin, J. M. Hellerstein, W. Hong, S. Krishnamurthy, S. R. Madden, V. Raman, F. Reiss, M. A. Shah, "TelegraphCQ: Continuous Dataflow Procesing for an Uncertain World," Proceedings of the CIDR conference, 2003.

[6] S. Nittel, A. Stefanidis, I. Cruz, M. Egenhofer, D. Goldin, A. Howard, A. Labrinidis, S. Madden, A. Voisard, M. Worboys, "Report from the First Workshop on GeoSensor Networks," SIGMOD record, Special Issue on "Sensor Network Technology Infrastructure, Security, Data processing, and Deployment," Ed.Vijay Kumar, 2004.

[7] B. Babcock, S. Babu, M. Datar, R. Motwani, J. Widom, "Models and Issues in Data Stream Systems," In Proc. of the ACM Symposium on Principles of Database Systems, 2002, pp. 1-16.

[8] S. Chandrasekaran, M. J. Franklin, "Streaming queries over streaming data," In Proc. of the VLDB Conference, 2002.

[9] J. Kang, J. F. Naughton, S. D. Viglas, "Evaluating window joins over unbounded streams," In ICDE, 2003.

[10] S. Madden, M. Shah, J. Hellerstein, V. Raman, "Continuously adaptive continuous queries over streams," In Proc. of the SIGMOD Conference, 2002.

[11] M. A. Hammad, W. G. Aref, A. K. Elmagarmid, "Stream window join: Tracking moving objects in sensornetwork databases," In SSDBM, 2003.

[12] T. Urhan, M. Franklin, "XJoin: A reactively scheduled pipelined join operator,"" IEEE Data Engineering Bulletin, 2000.

[13] 임효상, 이재길, 이민재, 황규영, "데이터와 질의의 이원성을 이용한 데이터스트림에서의 연속질의 처리," 정보과학회논문지, 제33권 제3호, 2006.

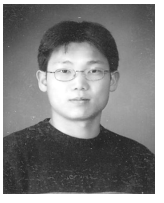
[14] A. Corral, M. Vassilakopoulos, Y. Manolopoulos, "Algorithms for joining R-trees and linear region quadtrees," In Advances in Spatial Databases--6th International Symposium, Lecture Notes in Computer Science, vol. 1651, pp. 251 - 269.

[15] G. R. Hjaltason, H. Samet, "Improved bulk-loading algorithms for quadtrees" In Proceedings of the 7th ACM International Symposium on Advances in Geographic Information Systems, pp. 110-115.

[16] M.-L. LO, C. V. Ravishankar, "Spatial joins using seeded trees," In Proceedings of the ACM SIGMOD Conference, pp. 209 - 220.

[17] 김진덕, 홍봉희, "고정 그리드를 이용한 병렬 공간 조인을 위한 비용 모델," 정보과학회논문지, 제28권 제4호, 2001.

[18] Y. Theodoridis, E. Stefanakis, T. Sellis, "Cost Models for Join Queries in Spatial Databases," Proc. of Int. Conf. on Data Engineering, 1998, pp. 476-483.



유 기 현

2007년 군산대학교 컴퓨터정보공학과 졸업 (학사)

2007년~현재 군산대학교 컴퓨터정보공학과 석사과정.

2008년~현재 해양연구원 위촉연구원

관심분야는 데이터베이스, GIS, 데이터스트림

남 광 우

1995년 충북대학교 전자계산학과 졸업 (학사)
1997년 충북대학교 전자계산학과 졸업 (석사)
2001년 충북대학교 전자계산학과 졸업 (박사)
2001년~2004년 한국전자통신연구원 텔레매틱스연구단
2004년~현재 군산대학교 컴퓨터정보공학과 조교수
관심분야는 데이터베이스, GIS, LBS 정책 및 기술, 데이터스트림, 지오 센서네트워크