

센서 네트워크에서 비트 벡터를 이용한 효율적인 레코드 필터링 인-네트워크 조인 전략

송 임 영*, 김 경 창**

Efficient Record Filtering In-network Join Strategy using Bit-Vector in Sensor Networks

Im-Young Song*, Kyung-Chang Kim**

요 약

본 논문에서는 센서 네트워크에서 효율적인 조인 처리를 수행하여 통신비용을 최소화할 수 있는 인-네트워크 조인 전략인 RFB(Record Filtering using Bit-vector) 조인 전략을 제안하였다. 제안하는 조인 전략은 통신비용을 감소하기 위하여 데이터의 크기를 줄이기 위해 비트 벡터(Bit-vector)를 이용한다. 또한, 센서 네트워크의 특성상 조인을 위해 중앙의 서버로 모든 데이터를 전송하기가 어렵기 때문에 본 논문의 조인 전략은 조인 처리의 이른 단계에서 조인 결과에 해당되지 않는 데이터를 제거하여 불필요한 통신비용을 감소시켜 통신비용을 최소화하기 위한 네트워크 내에서 효율적인 조인 질의 수행 전략이다. 본 논문에서는 시뮬레이션을 통하여 기존의 대표적인 조인 전략인 시놉시스를 전송하여 조인을 수행하는 시놉시스 조인(Synopsis Join, SNJ)과 비교하여 통신비용 효율적임을 보였다.

Abstract

The paper proposes RFB(Record Filtering using Bit-vector) join algorithm, an in-network strategy that uses bit-vector to drastically reduce the size of data and hence the communication cost. In addition, by eliminating data not involved in join result prior to actual join, communication cost can be minimized since not all data need to be moved to the join nodes. The simulation result shows that the proposed RFB algorithm significantly reduces the number of bytes to be moved to join nodes compared to the popular synopsis join(SNJ) algorithm.

▶ Keyword : 센서 네트워크(Sensor Network), 인-네트워크 조인(in-network join), 조인 전략(join strategy), 통신비용(communication cost), Synopsis 조인(Synopsis join), Semi 조인(Semi join), PERF 조인(PERF join), 비트 벡터(Bit-vector)

• 제1저자 : 송임영 교신저자 : 김경창

• 투고일 : 2010. 03. 08, 심사일 : 2010. 03. 10, 게재확정일 : 2010. 04. 08.

* 홍익대학교 컴퓨터공학과 재학 ** 홍익대학교정보컴퓨터공학부 교수

※ 이 논문은 2009년도 정부(교육과학기술부)의 지원으로 한국연구재단의 지원을 받아 수행된 기초연구사업임(No. 2009-0076875)

I. 서론

센서 네트워크는 각 센서 노드가 데이터 스트림을 생성하는 분산 데이터베이스 시스템으로 볼 수 있다. 이와 관련하여 오늘날 많은 연구 문헌[1,2,3]에서는 센서 데이터베이스라는 용어가 널리 사용되고 있다. 결과적으로 센서 네트워크에 존재하는 데이터들은 특정 형식과 의미를 갖는 데이터 레코드들이며, 관계형 데이터베이스 테이블로 모델링할 수 있으며, 센서 네트워크를 사용하는 어떠한 분야든지 결국 데이터를 수집하여 처리하는 것을 목표로 한다는 점에서 맥락을 같이한다.[4,5] 데이터베이스에서와 마찬가지로 감지된 센서 데이터 레코드를 수집하고 처리하기 위해 센서 네트워크에 질의를 보내는데, 센서 노드들은 제한적인 배터리 에너지를 갖기 때문에 센서 네트워크에서 분산 SQL 질의를 구현하기 위해서는 배터리 에너지의 주요 소비원인 통신비용을 최소화해야 한다. 통신비용을 최소화하기 위해서는 조인을 수행하는 사이트를 외부 데이터베이스 서버가 아닌 센서 네트워크 내의 노드들로 지정하여야 한다. 그래서 모든 데이터를 외부 데이터베이스 서버로 보내지 않고 센서 네트워크 내[6,7]에서 데이터베이스 질의를 구현하는 방법이 배터리 에너지를 절약하는 근본적인 방법이다.

본 논문에서는 센서 네트워크에서 효율적인 조인 처리를 수행하여 통신비용을 최소화할 수 있는 인-네트워크 조인 전략인 RFB(Record Filtering using Bit-vector) 조인 전략을 제안하였다. 제안하는 조인 전략은 센서 네트워크 조인 전략과 통신비용 효율적인 분산 데이터베이스의 조인 전략을 통합하였다. 통신비용을 줄이기 위해 중앙의 서버로 모든 데이터를 전송하지 않고 조인에 필요한 최소한의 데이터와 비트 벡터(Bit-vector)를 전송하여 조인 처리의 이른 단계에서 조인 결과에 해당되지 않는 데이터를 제거하여 통신비용을 최소화한다. 그리고 조인을 위해 센서 네트워크 내에서 많은 데이터의 이동이 이루어져야 하며 조인을 수행하는 지정 센서 노드에 많은 부하가 걸릴 수 있고 장애가 발생하면 조인 자체를 수행할 수 없다는 단점을 해결하기 위해 조인 연산 수행을 센서 네트워크 내의 여러 센서 노드들을 지정하여 분산 수행한다. 본 논문에서는 제안하는 조인 전략과 성능 평가를 위하여 센서 네트워크의 대표적인 조인 전략인 시놉시스 조인(Synopsis Join 이하 SNJ)[8]과 비교하였다.

본 논문의 구성은 다음과 같다. 2장에서는 센서 네트워크에서의 조인 전략 및 일반적인 조인 전략에 대하여 설명하고 분산 데이터베이스 조인 전략에 대하여 설명한다. 3장에서는

RFB의 소개 및 조인 전략의 세부사항을 설명한다. 4장에서는 RFB의 예를 설명하고 마지막 5장에서는 실험을 통해서 각 조인 전략을 비교하고, 통신비용을 그래프를 통한 가시적 성능 분석을 한다. 마지막으로 6장에서는 본 논문의 연구 내용을 정리 및 요약한다.

II. 관련 연구

1. 센서 네트워크 조인 전략

1.1 센서 네트워크에서 조인 전략

다양한 센서로부터 다양한 데이터가 수집되지만, 하나의 센서로부터 수집되는 데이터는 그 처리 능력으로 인해 정보의 종류가 한정적이다. 따라서 종합적인 정보를 얻고자 할 때, 특정 시간이나 위치를 기반으로 조인 연산을 수행하게 된다.

조인 질의를 구현하기 위해서는 먼저 어떤 조인 질의를 사용할 것인지를 결정해야 하는데 본 논문에서는 이진 동등-조인(Binary equi-join 혹은 BEJ) 질의를 고려한다.

조인의 예를 들기 위해 하나의 도로망을 덮는 센서 네트워크를 고려하자. 각 센서 노드는 자동차의 아디를 감지하고, 자동차의 감지된 시간 timestamp를 기록하고 그 기록들을 고정 기간(예를 들어 1시간) 동안 보관한다고 가정하자. 또한 N_R 과 N_S 가 각각 지역 Region1과 지역 Region2에 대한 센서 노드의 집합이며 테이블 R과 S는 각각 센서 노드 집합인 N_R 과 N_S 에 저장된다고 가정하자. 두 지역 사이를 이동하는 자동차들의 속도를 결정하기 위한 조인 질의는 다음과 같다.

```
SELECT R.autoId, R.time, S.time
FROM R, S
WHERE R.loc IN Region1 AND S.loc IN Region2
AND R.autoId = S.autoId
```

위의 조인 질의를 처리하기 위해서는 N_R 과 N_S 로부터 측정된 센서 값들을 수집하고 속성 autoId에 대해 조인해야 한다.

센서 네트워크의 일반적인 조인 전략은 조인 지역의 위치에 따라 Naive join, Sequential join, Centroid join 등 조인 전략을 분류할 수 있다.[9] 이런 일반적인 조인 전략의 주 문제점은 낮은 조인 선택도의 질의에 대한 통신 오버헤드이다. 이것은 테이블 R과 S의 조인 후보 튜플이 아닌 튜플들이 조인 지역 F로 전송되는 것이다.

낮은 조인 선택도의 질의에 대한 통신 오버헤드를 해결하기 위하여 제안된 SNJ는 최종 조인을 위해 전송되는 테이블 R과 S의 튜플 수를 줄이며 감지된 데이터의 시놉시스를 사용하여 조인 결과와 관련이 없는 데이터를 제거하는 것이다. 시놉시스의 의미는 조인을 수행하기 위한 테이블의 요약이며 따라서 해당 테이블 보다는 크기가 작으며 각 센서는 자체 로컬 테이블에 대한 시놉시스를 생성한다.

시놉시스 조인 전략은 3단계로 이루어진다. 1단계는 시놉시스 조인, 2단계는 통보(notification) 전송, 그리고 3단계는 최종 조인이다. 1단계인 시놉시스 조인은 다음과 같이 수행된다. 각 노드 $n_i \in N_R$ 는 테이블 R의 일부인 로컬 테이블 R_i 를 저장한다. 각 노드 n_i 는 R_i 의 조인 속성 A_i 를 추출하고 A_i 의 중복된 값들의 빈도수를 계산함으로써 로컬 시놉시스 $S(R_i)$ 를 생성한다. 테이블 R과 S의 시놉시스를 조인하여 최종 조인 후보 튜플들을 결정하기 위하여 시놉시스 조인 지역 L에 있는 시놉시스 조인 노드의 집합인 N_L 이 선택된다. 시놉시스 조인 노드들은 N_R 과 N_S 노드들로부터 시놉시스가 도착하면 시놉시스 조인을 수행한다. 시놉시스 조인 전략의 2단계인 통보 전송에서는 N_R 과 N_S 에 있는 각 센서 노드에게 최종 조인 후보 튜플들을 통보해야 한다. 이를 위해 시놉시스 조인 노드 n_i 는 로컬 시놉시스의 근원인 센서의 식별자 아이디를 저장한다. 또한 각 조인 속성 값 a에 대해서 조인 속성 a값을 갖는 튜플들을 저장하는 두 개의 센서 집합인 N_R^a 와 N_S^a 를 식별하고 이들 튜플들을 조인하는 최종 조인 노드 $n_f \in N_f$ 를 선택한다. 시놉시스 조인 전략의 3단계인 최종 조인에서는 시놉시스 조인 노드 n_f 로부터 통보 메시지를 받은 N_R 혹은 N_S 의 각 노드는 함께 통보 받은 최종 조인 노드 n_f 에 조인 속성값 v에 해당하는 후보 튜플들을 보낸다. 최종 조인 노드 n_f 는 $R^v \bowtie S^v$ 조인을 수행하고 조인 결과를 쿼리 싱크로 보낸다.

2. 분산 데이터베이스에서 조인 전략

본 논문에서 제안하는 통신비용을 최소화하는 조인 전략에서 이용되는 기존의 분산 데이터베이스를 위한 대표적인 조인 전략들은 Semi 조인[10], Bloom 조인[11] 그리고 PERF 조인[12]이 있다.

먼저 테이블 R에서 테이블 S로의 Semi 조인 ($S \times R$) 전략은 다음과 같다. 테이블 R과 S의 조인 속성들에 대해 테이블 R을 프로젝션(projection)하여 새 테이블 P_R 을 얻는다. P_R 을 테이블 S가 있는 사이트로 이동시켜 테이블 P_R 과 테이블 S의 조인을 수행하여 새 테이블 S' 을 얻는다. 결과 S' 는 원래 테이블 S보다는 훨씬 적다.

PERF 조인은 양방향 Semi 조인의 백워드(backward) 단계

의 비용을 최소화 하도록 설계되었다. 양방향 semi 조인에서 처럼 테이블 S는 테이블 P_R 과 Semi 조인함으로써 크기를 줄인다. 하지만 P_S 을 R에 전송하는 것 대신에 P_R 의 각 튜플에 대해 1개의 비트를 포함하는 비트 벡터를 보내는데 그 비트가 P_S 에 있으면 1로 아니면 0으로 세트된다. 비트 벡터의 비트 순서는 P_R 과 같은 튜플 순서이다. PERF는 원래의 P_S 보다 훨씬 적다.

III. RFB(Record Filtering using Bit-vector) 전략

이번 장에서는 본 논문에서 제안하는 통신비용을 최소화할 수 있는 인-네트워크 RFB(Record Filtering using Bit-vector) 조인 전략을 설명한다. RFB 조인 전략을 기술하는데 쓰이는 용어들은 다음과 같다. 이 용어들을 논문 전반에 걸쳐 사용하도록 하며 통신비용을 산출하기 위해 필요한 값들은 [8]에서 사용한 값들을 본 논문에서 제안하는 조인 전략에 맞게 적용하였다.

- 두 지역 Region1과 Region2의 센서들의 집합은 각각 N_R 과 N_S 이며 각 지역에서 센싱된 데이터는 네트워크에 저장된 두 개의 분산 테이블 R과 S이다.
- P_R : R의 세미 테이블이며, $|P_R|$ 는 튜플 수이다.
- P_S : S의 세미 테이블이며, $|P_S|$ 는 튜플 수이다.
- P_{R^v} : P_R 에서 최종조인에 참여하는 속성만 가지는 세미 테이블이다.
- P_{S^v} : P_S 에서 최종조인에 참여하는 속성만 가지는 세미 테이블이다.
- Bit-vector(R) : P_R 의 각 튜플에 대해 1개의 비트를 포함하는 비트 벡터이다.
- Bit-vector(S) : P_S 의 각 튜플에 대해 1개의 비트를 포함하는 비트 벡터이다.
- c_h : 세미 테이블 조인 노드이다.
- N_f : 최종 조인 지역이며, $|N_f|$ 는 노드 수이다.
- R^v : 조인 속성 v에 대한 R의 조인 후보 튜플 집합이다.
- S^v : 조인 속성 v에 대한 S의 조인 후보 튜플 집합이다.

RFB의 조인 전략 단계는 다음과 같다.

1. 세미 테이블 생성

1.1 조인 속성 값들을 사용하여 Region1과 Region2의 세

- 미 테이블을 생성한다.
- 2. 세미 테이블 조인
 - 2.1 각 지역의 P_R 과 P_S 를 c_H 로 전송한다.
 - 2.2 P_R 과 P_S 조인 후 각 지역으로 통보할 비트 벡터 생성한다.
 - 2.3 Bit-vector(R), Bit-vector(S)를 각각 R과 S에 통보 전송한다.
- 3. 최종 조인
 - 3.1 두 지역의 조인 후보 튜플들을 최종 조인 지역 N_F 로 전송하여 최종 조인을 한다.
 - 3.2 조인 결과를 쿼리 싱크로 전송한다.

그림 1 은 RFB 조인 전략의 전체 흐름이다.

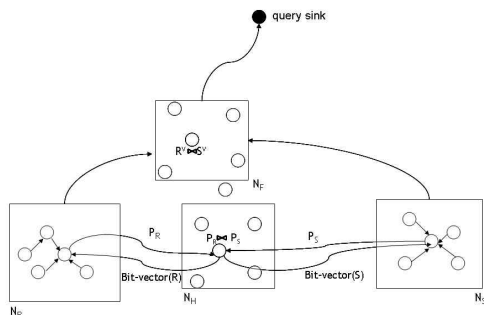


그림 1. RFB 조인 전략
Fig. 1. RFB Join strategy

위와 같은 조인 전략 단계를 가지는 RFB의 세부 사항에 대하여 이어지는 절들에서 구체적으로 설명한다.

1. 조인 속성값들을 사용하여 세미 테이블 생성

쿼리 싱크는 조인을 수행하고자하는 Region1과 Region2의 중심에 해당하는 지역을 계산하여 그 지역에 가까운 노드를 두 지역의 중심 노드 c_r 과 c_s 로 선택한다. 쿼리 싱크는 generalized fermat's point[13]를 이용하여 c_r , c_s , 쿼리 싱크로 이루어지는 삼각형 합 of 최소점인 g 를 계산하고 g 에 가까운 노드와 질의 Q를 라우팅 프로토콜 GPSR[14]를 이용하여 각 지역의 중심 노드에 전송한다. g 는 R과 S의 후보 튜플을 조인할 N_F 지역의 라우팅 노드로 이용될 노드이다. c_r 과 c_s 는 질의 Q를 각 지역의 이웃 노드들에게 전파하고 각 센서 노드는 질의를 받으면 로컬 세미 테이블을 생성한다.

각 지역의 중심 노드 c_r 과 c_s 에서 생성한 P_R , P_S 를 조인 속성을 기준으로 오름차순으로 정렬하여 c_H 로 전송한다.

2. 세미 테이블 조인

세미 테이블 조인 단계에서는 R과 S지역의 세미 테이블 P_R 과 P_S 를 조인하여 Bit-vector(R)과 Bit-vector(S)를 생성하는 단계이다.

2.1 각 지역의 세미 테이블을 c_H 로 전송

네트워크 내에서 R, S, g 그리고 쿼리 싱크 노드를 제외하고 모든 센서들로부터 세미 테이블 조인 노드 c_H 를 선택해야 한다. c_H 의 위치는 지역 세미 테이블 P_R , P_S 의 라우팅 통신비용을 최소화 할 수 있는 노드로 선택한다. Region1과 Region2의 데이터 분배와 네트워크 토폴로지를 이용하여 최적의 세미 테이블 노드를 찾는다.

세미 테이블 노드의 선택을 단순화하기 위해, c_r 과 c_s 를 연결한 선이 지나는 정사각형 N_H 를 선택한다. N_H 의 중심 c_H 는 다음 식으로 결정된다. Region1의 범위는 $area(R)$ 로 표시하고 Region2의 범위는 $area(S)$ 로 표시한다.

$$center(N_H) = c_r \cdot (1 - f) + c_s \cdot f, \dots\dots\dots (1)$$

$$f = \frac{area(S) |P_S|}{area(S) |P_S| + area(R) |P_R|}$$

c_r , c_s 는 N_H 의 사이즈와 중심 노드 c_H 를 계산하여, P_R 과 P_S 를 c_H 로 전송한다.

2.2 세미 테이블 조인 후 비트 벡터 생성

테이블 R과 S의 조인 후보 속성을 결정하기 위해 P_R 과 P_S 를 조인한다. 조인 수행 후 P_R 과 P_S 의 각 튜플에 대해 1개의 비트를 포함하는 비트 벡터 컬럼을 추가한다. 속성값이 조인 후보 속성에 속하면 그 비트가 1로 아니면 0으로 세트된다. 비트 벡터의 비트 순서는 P_R 과 P_S 와 같은 튜플 순서여야 하므로, c_H 에서 조인이 수행되었다면 조인 속성을 기준으로 정렬하여 비트 벡터 컬럼 Bit-vector(R), Bit-vector(S)만 각각 c_r 과 c_s 에 전송한다. 만약, 여러 노드에 걸쳐 조인이 이루어졌다면 N_H 의 중심 노드에서 합쳐서 최종 비트 벡터를 생성 후 조인 속성을 기준으로 오름차순 정렬 후 비트 벡터 컬럼만 각 지역으로 전송한다.

2.3 통보 전송

최종 조인을 위하여 N_F 로 전송할 R과 S의 조인 후보 튜플을 선택하기 위하여 c_r 과 c_s 에 각각 Bit-vector(R), Bit-vector(S)를 전송한다.

c_r 은 전송받은 Bit-vector(R)를 P_R 과 비교하여 조인 후보 속성에 해당하는 튜플을 각 노드 $n_i \in N_R$ 에 뿌려준다. N_S 노드들도 Bit-vector(S)와 비교하여 최종 조인 후보 튜플들을 선택할 수 있다. N_R, N_S 노드의 최종 조인 후보 튜플들을 쿼리 싱크가 질의 Q와 함께 전송한 중심점 g 로 전송한다.

3. 최종 조인

테이블 R과 S의 모든 조인 후보 튜플을 g 로 전송하여 R^v 와 S^v 의 조인을 수행하고 조인 결과를 쿼리 싱크로 전송한다.

g 는 조인 속성 v 에 대한 최종 조인 노드 N_F 의 라우터로 이용한다. 조인 속성 v 에 대한 최종 조인 노드 $n_f(v)$ 는 g 에서 제시하여 g 주변 노드로 결정하고 g 는 R과 S의 조인 후보 튜플 수를 통보받고 필요 노드 수를 결정한다. $|C(R_i)|$ 은 각 노드 $n_i \in N_R$ 로부터 전송되는 조인 후보 튜플 수이며, $|C(S_j)|$ 는 각 노드 $n_j \in N_S$ 로부터 전송되는 조인 후보 튜플 수이다.

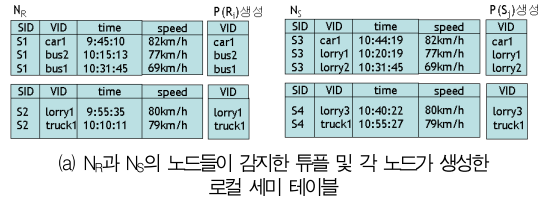
$$|N_F| = (\sum_{n_i \in N_R} |C(R_i)| + \sum_{n_j \in N_S} |C(S_j)|) / m \dots\dots\dots (2)$$

N_F 의 지리적 정보를 받으면 g 는 N_R 과 N_S 에서 전송되는 조인 후보 튜플들의 속성 값 v 를 기반으로 $|N_F|$ 그리드로 해시하여 해당 n_f 로 튜플을 전송한다. 하나의 센서 $n_i \in N_R$ 에 대해 $C(R_i)$ 는 $|N_F|$ 개로 구성된다. 각 그리드 g 는 $C(R_i)$ 의 부분으로 생성되었다. 만약, 조인 속성 v 에 대해 $n_f(v)$ 가 충분한 메모리를 가지지 못할 때는 이웃 노드들에게 조인 연산 수행 도움을 요청한다. N_R 과 N_S 로부터 튜플을 모두 전송 받으면 조인 수행 후 결과를 쿼리 싱크로 전송한다.

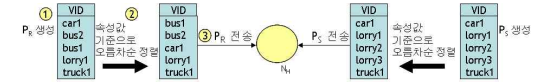
IV. RFB 조인 전략 예

이 장에서는 RFB 조인 전략의 예를 설명한다.

예를 간단히 설명하기 위하여 N_R 과 N_S 가 각각 Region1과 Region2에 대한 센서 노드의 세트이며, N_R 에는 $s1$ 과 $s2$ 두 개의 센서 노드가 있고 N_S 에는 $s3$ 와 $s4$ 두 개의 센서가 있다고 가정하였으며, 각 센서 노드들이 감지한 튜플들은 SID(노드 아이디), VID(차량 아이디), 감지된 시간(time), 차량속도(speed) 등의 속성값을 가진다. 쿼리 싱크로부터 질의를 전송 받은 후 N_R 과 N_S 의 각 노드는 그림 2 (a)와 같이 로컬 세미 테이블 $P(R_i)$ 와 $P(S_j)$ 를 생성한다. 다음으로 각 노드에서 생성한 로컬 세미 테이블을 c_r 과 c_s 에서 중복 조인 속성을 제거하고 오름차순으로 정렬 후 세미 테이블 조인 지역 o_f 로 전송한다.



(a) N_R 과 N_S 의 노드들이 감지한 튜플 및 각 노드가 생성한 로컬 세미 테이블

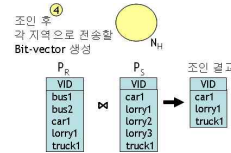


(b) 세미 테이블을 속성값 기준으로 정렬 후 o_f 로 전송

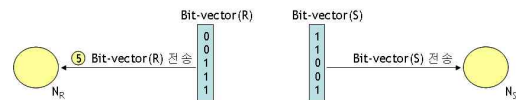
그림 2 세미 테이블 생성 및 o_f 로 전송

Fig. 2. Semi table generation and transmission to o_f

o_f 로 P_R 과 P_S 가 전송되면 그림 3의 (a)는 두 세미 테이블을 조인하여 최종 조인 지역으로 전송할 조인 속성을 알 수 있으며, 각 지역으로 통보할 Bit-vector를 생성한다. N_R 과 N_S 로 P_R 과 P_S 를 전송하는 대신 P_R 과 P_S 의 각 튜플에 대해 1개의 비트를 포함하는 Bit-vector를 보내는데 그 비트가 조인 결과에 있는 조인 속성값이면 1로 아니면 0으로 세트된다. 그림 3의 (b)는 생성한 Bit-vector(R)과 Bit-vector(S)를 각 지역으로 통보 전송하는 단계이다.



(a) o_f 에서 P_R 과 P_S 조인



(b) Bit-vector 생성 후 각 지역으로 전송

그림 3. Bit-vector 생성 및 통보 전송

Fig. 3. Bit-vector generation and Notification transmission

각 지역으로 전송된 Bit-vector의 비트 순서는 P_R, P_S 와 같은 튜플 순서이므로 그림 4에서 c_r 과 c_s 는 전송받은 Bit-vector와 P_R, P_S 를 비교하여 조인 후보 속성들을 선택하여 해당 조인 속성값에 대한 정보를 지역의 다른 노드들에게 뿌려준다. 각 노드들은 조인 후보 속성값을 가지는 튜플을 선택하여 최종 조인 노드로 전송하며 각 노드 $n_f \in N_F$ 는 같은 조인 값 범위 v 를 갖는 두 개의 서브 테이블 R^v 와 S^v 를 조인하여 조인 결과를 쿼리 싱크로 전송한다.

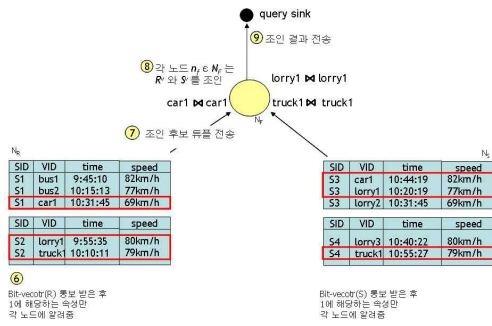


그림 4. 튜플 선택 및 조인 후보 튜플 전송
Fig. 4. Tuples selection and Candidate tuples transmission

제안하는 RFB는 배터리 에너지의 주요 소비원에 해당하는 통신비용을 줄이기 위하여 기존 SNJ와 비교하여 N_R 과 N_S 의 로컬 시뮬시스를 생성할 필요가 없으며 대신 각 지역에서 조인 속성을 프로젝트한 P_R 과 P_S 를 생성하여 세미 테이블 조인 노드로 전송한다. SNJ의 시뮬시스 테이블은 조인 속성, 노드 아이디, 조인 속성 중복 횟수와 튜플 번호 등 네 가지 속성을 가지는 튜플들로 이루어졌지만 RFB는 조인 속성만 프로젝트하고, 해당 지역에 센싱된 튜플의 속성 여부만 알면 되므로 각 지역의 노드들이 전송하는 조인 속성값의 중복을 제거하여 세미 테이블을 생성하므로 세미 테이블 조인 노드로 보내는 테이블의 속성 수를 줄여 통신비용이 SNJ에 비해 효율적이다. 세미 테이블 조인 수행 후 최종 조인에 참여하는 조인 속성을 각 지역에 통보하기 위하여 SNJ는 시뮬시스 테이블의 노드 아이디를 참조하여 각 지역의 노드들에게 통보 전송하는데 통보 전송시 N_R 과 N_S 의 해당 조인 속성의 카운트 값과 최종 조인 노드의 위치를 계산한 컬럼을 함께 전송한다. 이에 비해 RFB는 P_R 과 P_S 의 각 튜플에 대해 1개의 비트를 포함하여 조인 후보 속성에 속하면 1로 아니면 0으로 세트되어진 비트 벡터를 전송하여 SNJ보다 적은 비트 리스트를 이용하여 통보 전송함으로써 최소한의 데이터로 조인에 참여하는 튜플을 제거할 수 있어 SNJ에 비해 통신비용 효율적이다.

V. 실험

본 논문에서는 컴퓨터 시뮬레이션을 통해서 성능분석을 수행하였으며, 제안하는 조인 전략을 테스트하고 객관적인 효율성을 검증하기 위해 SNJ 조인 전략과 총 통신비용을 비교 분석하였다. 조인 전략의 주요 성능 기준은 조인을 구현할 때 발생하는 총 통신비용이다. 센서 네트워크에서 총 통신비용의 정의는 질의 배포부터 최종 조인 결과 수집까지의 데이터 전

송으로 본 논문에서의 총 통신비용이란 바이트 값으로 정량화한 전송 바이트를 의미한다.

1. 실험 환경

본 논문의 실험환경은 논문에서 제안하는 조인 전략들과 성능 비교를 하는 SNJ[8]의 실험 환경의 조건과 같게 시뮬레이션 하였다. 본 실험에서는 가로 세로 100×100 그리드에 균등하게 10,000개의 센서 노드로 배치된 센서 네트워크 환경을 시뮬레이션 하였다. 각 그리드의 중심에 하나의 센서 노드가 위치하고 쿼리 싱크는 중앙 위에 위치한다. 각 지역 Region1과 Region2는 오른쪽 아래와 왼쪽 아래에 위치해 있고 각 지역은 각각 800개의 센서 노드들이 존재한다. 테이블 R은 2000개의 튜플이 구성되었고, 테이블 S도 1000개의 튜플로 구성되었다. Region1과 Region2의 튜플들은 각 센서노드들에 균등하게 분포되어 있다. 소스 노드에서 최종 노드로 하나의 메시지를 라우트하는데 요구되는 홉수는 두 센서 노드 사이의 거리와 통신 반경을 사용한다. 네트워크 트래픽 분석을 단순화하기 위해 메시지 전송시 메시지 전송 실패는 없다고 가정하였다. 튜플 사이즈는 40바이트이고 SNJ에 사용한 시뮬시스 사이징은 10바이트이다.

이러한 절제에서는 앞에서 설명한 실험 환경을 바탕으로 각 조인 전략 별로 측정된 총 통신비용을 비교 분석한다. 이때, Region1과 Region2의 튜플 생성시 조인 속성은 1과 100사이의 랜덤값을 가지도록 생성하기 때문에 실험을 반복할 때마다 측정값이 달라질 수 있다. 즉, 생성되는 속성값에 따라 조인 조건을 만족하는 튜플수가 변화하여 각 알고리즘별 총 통신비용의 차이가 발생할 수 있다. 이러한 변화를 반영하기 위해서 모든 실험의 최종 측정값은 동일한 파라미터 값들에 대해 5번씩 실험한 결과값들의 평균으로 정하였다.

2. 실험 결과 및 분석

2.1 조인 선택율에 대한 성능

조인 선택율에 따른 전송 바이트를 확인할 수 있다. 단, 메모리 용량은 250×40 바이트, 통신 반경은 4 유닛(units)으로 고정시켰다. RFB의 총 통신비용은 각 지역으로부터 c_1 로 전송되는 세미 테이블의 통신비용, 각 지역으로 통보되는 전송 비용과 N_R 과 N_S 에 조인 후보 속성에 해당하는 튜플을 뿌려주는 비용, 조인 후보 속성을 최종 조인 노드에 전송하는 비용 그리고 조인 결과를 쿼리 싱크에 전송하는 비용을 합한 것이다. 먼저, c_1 로부터 c_2 로 전송되는 P_R 과 c_3 로부터 c_4 로 전송되는 P_S 의 통신비용은 다음과 같다.

$$|P_R| \text{dist}(c_r, c_h) + |P_S| \text{dist}(c_s, c_h) \dots\dots\dots(3)$$

c_r 은 전송받은 Bit-vector(R)를 P_R 과 비교한 후, 조인 후보 속성에 해당하는 튜플을 각 노드 $n_i \in N_R$ 에 뿌려준다. 통신비용은 다음과 같다. $|C(P_R)|$ 은 각 노드 $n_i \in N_R$ 로 전송되는 조인 후보 속성이며, $|B_i(P_R)|$ 는 P_R 의 비트 벡터이다. S 지역도 마찬가지다.

$$|B_h(P_R)| \text{dist}(c_h, c_r) + \sum_{n_i \in N_R} |C(P_R)| \text{dist}(c_r, n_i) \dots\dots\dots(4)$$

조인 후보 튜플이 최종 조인 수행 후 조인 결과가 쿼리 싱크에 전송되는 비용은 N_R^V 과 N_S^V 에서 g 까지 조인 속성 값 v 와 함께 조인 후보 튜플을 보내는 통신비용과 g 에서 해싱되어 속성 값에 따라 $n_i(v)$ 로 전송되는 비용, 그리고 $n_i(v)$ 에서 조인 수행 후 싱크까지 결과를 보내는 비용으로 결정된다.

N_R^V 과 N_S^V 는 각 지역의 노드에서 조인 속성 값 v 를 포함하는 노드를 의미하고, $|R^V|$ 와 $|S^V|$ 는 조인 속성 값 v 와 함께 n_i 의 R 튜플 수와 n_i 의 S 튜플 수를 의미하며, $|R^V|$ 는 N_R 에서 조인 후보 속성 v 를 가진 튜플들의 합이며, $|S^V|$ 는 N_S 에서 조인 후보 속성 v 를 가진 튜플들의 합이다.

$$\begin{aligned} & \sum_{n_i \in N_R^V} |R^V| \text{dist}(n_i, g) + \sum_{n_j \in N_S^V} |S^V| \text{dist}(n_j, g) \\ & + \sum_{n_f \in N_F} |R^V| \text{dist}(g, n_f) + \sum_{n_f \in N_F} |S^V| \text{dist}(g, n_f) \\ & + |R^V| \times |S^V| \text{dist}(n_f, \text{sink}) \dots\dots\dots(5) \end{aligned}$$

그림 5는 조인 선택율에 따른 전송 바이트를 나타낸다.

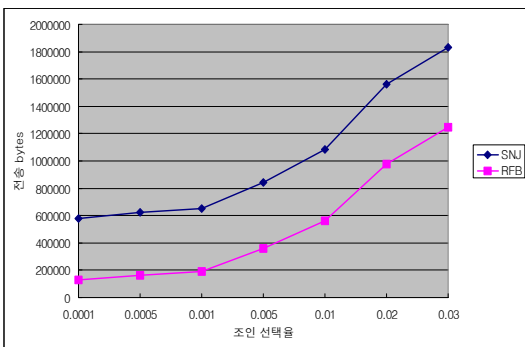


그림 5 조인 선택율에 따른 전송 바이트
Fig. 5. Impact of Selectivity

모든 조인 선택율에 대해 SNJ 보다 RFB가 전송 바이트가 현저히 적음을 확인할 수 있다. 조인 선택율이 0.01 보다 작을 때는 SNJ의 30% 이하에 해당하는 전송 바이트를 보였으며,

조인 선택율이 0.01 이상일 때는 RFB가 SNJ의 50-70%에 해당하는 전송 바이트를 보였다. 이는 조인 선택율이 높아지면 그만큼 조인 후보 속성값이 많아지며 각 지역의 중심 노드 c_r , c_s 에서 Bit-vector(R)과 Bit-vector(S)와 비교후 R과 S의 지역 노드들에게 전송하는 조인 후보 튜플수가 증가하여 각 지역 내 전송 바이트가 높아지기 때문이다.

그림 6은 N_R 과 c_r 사이의 하나의 노드가 조인 선택율에 따라 처리해야 하는 바이트이다. SNJ는 시놉시스 튜플과 R의 조인 후보 튜플을 전송하며, RFB는 P_R 과 Bit-vector(R)을 전송하는 노드들로 조인 전략에 따라 같은 지역의 노드지만 데이터 전송량이 큰 차이를 보이며 이것은 통신비용에 사용되는 배터리 소모량의 차이 역시 크다는 것을 확인할 수 있다.

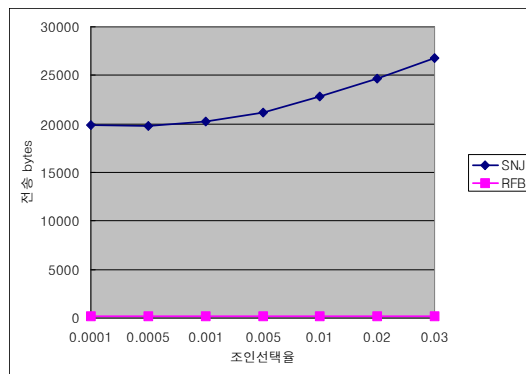


그림 6. N_R 과 c_r 사이의 전송 바이트
Fig. 6. Impact of Selectivity between N_R and c_r

모두 조인 선택율이 높아짐에 따라 SNJ는 하나의 노드에서 처리해야하는 바이트가 증가하지만 RFB는 조인 선택율에 상관없이 일정한 바이트를 처리하는 것을 확인할 수 있다. 이는 중복된 속성은 제거하고 세미 테이블을 생성하여 전송하고 RFB의 비트 벡터 사이즈는 속성값의 수와 같으므로 조인 선택율이 증가하여도 전송 바이트는 일정하였고, SNJ는 조인 선택율이 증가하면 N_R 과 N_S 로 뿌려지는 통보 메시지가 많아지므로 조인 선택율이 증가하면 그만큼 전송 바이트가 증가한다.

2.2 네트워크 밀도의 효과

네트워크 밀도를 다양하게 하여 조인 전략의 확장성을 확인하였다.

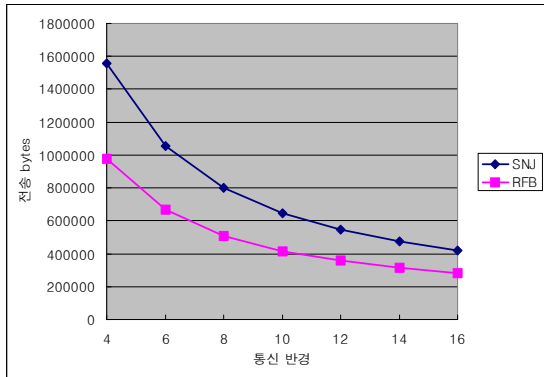


그림 7. 통신 반경에 대한 전송 바이트
Fig. 7. Impact of Network Density

그림 7는 통신 반경을 4 유닛에서 16 유닛까지 변경해가며 통신 반경에 대한 전송 바이트를 비교한 그래프이다. 이때, 메모리 용량은 250×40 바이트, 조인 선택율은 0.0001이다. 네트워크 밀도가 더 높아질수록 두 조인 전략에 대한 전체 통신비용은 감소한다. 이것은 큰 통신 범위와 네트워크 전역에 하나의 메시지를 보내는데 보다 소수의 홉을 필요로 하기 때문이다. 이처럼 통신 반경이 커짐에 따라 단위 데이터를 전송하기 위하여 필요로 하는 에너지도 증대될 것이므로 통신 반경에 따른 각 노드의 소모 에너지를 계산하였다. 모든 노드들의 초기 배터리 에너지는 2J로 동일하다고 가정하였으며 각 노드의 소모 에너지는 다음과 같이 계산하였다.

$$\text{전송} : E_{CHTX}(k, d) = (E_{elec} \times k) + (E_{amp} \times k \times d^2) \dots (6)$$

$$\text{수신} : E_{GWRX}(k) = E_{elec} \times k \dots (7)$$

노드별 소모 에너지를 산출하기 위해 필요한 각 요소들의 정의 및 해당 값들은 표 1에서와 같으며 [15]에서 사용한 값들을 동일하게 적용하였다.

표 1. 노드별 에너지 소모 요소
Table 1. Node energy consumption factor

항목	정의	값
E_{elec}	송수신회로 소모 에너지	50nJ/bit
E_{amp}	송신 앰프 소모 에너지	100pJ/bit/m ²
d	노드 전송 반경	4 유닛
k	송신 데이터 크기	

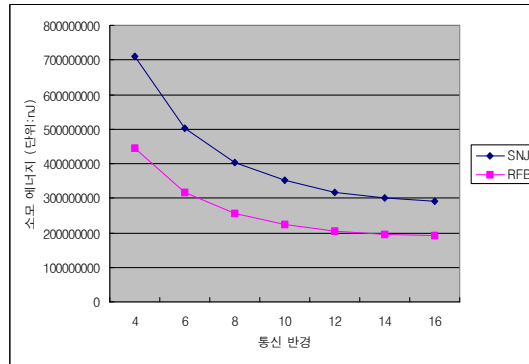


그림 8. 통신 반경에 대한 소모 에너지
Fig. 8. Energy consumption for communication radius

그림 8과 같이 통신 반경에 대한 각 노드별 소모 에너지를 계산한 결과 노드 전송 반경은 각 노드별 소모 에너지 계산에 별 영향을 주지 않고 전송 바이트가 줄어들어 따라 노드별 소모 에너지도 줄어들음을 확인할 수 있다.

2.3 메모리 용량의 효과

그림 9는 노드 메모리 용량의 변화에 따른 전송 바이트를 보여준다. 본 실험에서는 통신 범위는 4 유닛이며, 조인 선택율은 0.0001이다.

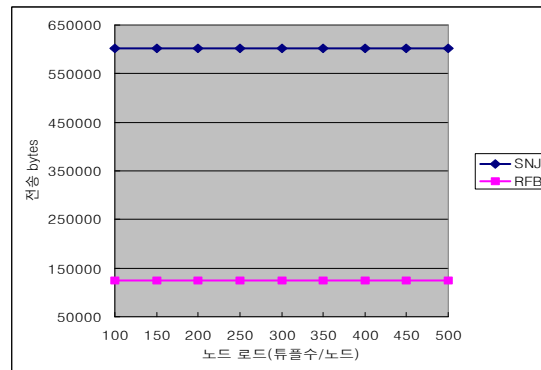


그림 9. 메모리 용량에 대한 전송 바이트
Fig. 9. Impact of Memory Capacity

두 조인 전략 모두 메모리 용량이 증가하여도 통신비용이 달라지지 않는다. 메모리 용량의 변화는 조인 노드들의 수에 영향을 준다. 메모리 용량이 클 경우 소수의 조인 노드들이 선택되고 쿼리 싱크에 조인 결과 튜플을 보내는 것에 대한 더 적은 전송 바이트를 요구한다.

VI. 결 론

센서 데이터베이스 분야에서 기존의 조인 처리 전략들은 통신비용을 줄이기 위해 최종 조인을 처리하는 조인 노드들을 어떻게 선택할지에 주요 관심을 두었다. 통신비용을 줄이기 위해 분산 데이터베이스 조인 전략들인 Semi 조인, Bloom 조인, PERF 조인 전략 등을 고려하는 연구 결과는 아주 미비한 실정이다.

본 논문에서는 센서 네트워크의 조인 전략과 통신비용 효율적인 분산 데이터베이스의 조인 전략들을 통합하여 센서 네트워크에서 효율적인 조인 처리를 수행하여 통신비용을 최소화할 수 있는 인-네트워크 조인 전략인 RFB(Record Filtering using Bit-vector)를 제안하였다. 제안하는 조인 전략은 통신비용을 감소하기 위하여 데이터의 크기를 줄이기 위해 비트 벡터를 이용하였다. 그리고 기존 데이터 대신 작은 크기의 비트 벡터를 전송함으로써 배터리 전력 소모를 줄이게 된다.

센서 네트워크의 대표적인 조인 전략인 SNJ 조인 전략과 비교하여 조인 속성 튜플을 가진 세미 테이블과 비트 벡터를 이용하여 조인 후보 속성을 통보하는 RFB는 전송 바이트가 현저히 줄어들었으며 SNJ와 비교하여 조인 후보 튜플을 최종 조인 노드에 전송하는 오버헤드도 없었다. 센서 노드 배터리 소모량도 SNJ에 비해 크게 줄었으며, 조인 선택율이 높아져도 하나의 노드에서 처리해야하는 바이트는 일정했다. 또한, 통신 환경에 대한 통신비용도 SNJ에 비해 훨씬 효과적이었으며, 노드 메모리 용량에 따른 전송 바이트도 RFB가 훨씬 적은 전송 바이트를 보였다. 즉, 제안한 조인 전략은 분산 데이터베이스의 조인 전략들을 이용하여 센서 네트워크 내에서 다량의 데이터 이동에 따른 통신비용을 대폭 줄일 수 있었다.

참고문헌

- [1] P. Bonnet, J. Gehrke, and P. Seshadri, "Towards Sensor Database Systems," ACM, vol. 1987, pp. 3-14, Jan. 2001.
- [2] J. Z. Sun, "An Energy-Efficient Query Processing Algorithm for Wireless Sensor Networks," LNCS, vol. 5061, pp. 373-385, June 2008.
- [3] Z. Zhang, X. Gao, X. Zhang, W. Wu, and H. Xiong, "Three Approximation Algorithms for Energy-Efficient Query Dissemination in Sensor Database System," LNCS, vol. 5690, pp.807-821, Aug. 2009.
- [4] 김회복, 신정훈, 김형진, "무선 센서 네트워크에서 정보 보호를 위한 키 분배 기법," 한국컴퓨터정보학회논문지, 제 14권, 제 6호, 51-57쪽, 2009년 6월.
- [5] 김용태, 정윤수, 박병주, 박길철, "무선 센서 네트워크 환경에 대한 센서 웹 서비스 프레임워크의 설계," 한국컴퓨터정보학회논문지, 제 14권, 제 10호, 123-131쪽, 2009년 10월.
- [6] A. Coman, and M. A. Nascimento, "A Distributed Algorithm for Joins in Sensor Networks," Proc. of Intl' Conference on SSDBM, 19th, pp. 27, July 2007.
- [7] A. Pandit, and H. Gupta, "Communication-Efficient Implementation of Range-Joins in Sensor Networks," Proc. of DASFAA, pp. 859-869, Mar. 2006.
- [8] H. Yu, E. Lim, and J. Zhang, "In-Network Join Processing for Sensor Networks," LNCS, vol. 3841, pp. 263-274, May 2006.
- [9] A. Coman, M. Nascimento, and J. Sander, "On Join Location in Sensor Networks," Proc. of MDM, pp. 190-197, May 2007.
- [10] P. A. Bernstein, and D. W. Chiu, "Using Semi-Joins to Solve Relational Queries," Journal of th ACM, 28-1, pp. 25-40, Jan. 1981.
- [11] A. Broder, and M. Mitzenmacher, "Network Applications of Bloom Filters : A Survey," Internet Mathematics, Vol. 1, pp. 485-509, Mar. 2004.
- [12] Z. Li, and K. Ross, "PERF Join : An Alternative to Two-way Semijoin and Bloomjoin," Proc. of Intl' Conference on Information and Knowledge Management, pp. 137-144, Nov. 1995.
- [13] I. Greenberg and R. A. Robertello, "The Three Factory Problem," Mathematics Magazine, 38-2, pp. 67-72, May 1965.
- [14] B. Karp and H. Kung. "GPSR: Greedy Perimeter Stateless Routing for Wireless Networks," In Proc. of ACM MobiCom '00, pp. 243 -254, July 2000.
- [15] W. B. Heinzelman, A. P. Chandarkasan, and H. Balakrishnan, "An Application-Specific Protocol Architecture for Wireless Microsensor Networks," Wireless Comm, IEEE, Vol 1, No. 4, pp. 660-670, Oct. 2002.

저자 소개



송 임 영

2002 : 서울산업대학교

전자계산학과 공학사.

2004 : 서울산업대학교

컴퓨터공학과 공학석사.

현 재 : 홍익대학교 컴퓨터공학과 재학

관심분야 : 객체지향DB,

Ubiquitous 환경에서의

Data Mining, 센서DB



김 경 창

1978 : 홍익대학교

전자계산학과 공학사.

1980 : 한국과학기술원

전산학과 공학석사.

1990 : University of Texas at

Austin 전산학과 공학박사

현 재 : 홍익대학교

정보컴퓨터공학부 교수

관심분야 : 객체지향DB, 주기억DB,

센서DB