

다차원 센서 데이터 저장의 효율적인 Scalability를 위한 비균등 영역 분할†

전상훈[○], 강홍구, 홍동숙, 박춘걸, 한기준
건국대학교 컴퓨터정보통신공학과

{shjeon[○], hkkang, dshong, cgpark, kjhan}@db.konkuk.ac.kr

Non-Equal Region Split for Efficient Scalability of Multi-dimensional Sensor Data Storage

Sang-Hun Jeon[○], Hong-Koo Kang, Dong-Suk Hong, Chun-Geol Park, Ki-Joon Han
Dept. of Computer Information & Communication Engineering, Konkuk University

요 약

데이터 중심 저장 방식의 센서 네트워크는 비슷한 값의 데이터를 같은 센서 노드에 저장한다. 따라서 센서 네트워크가 확장되고 비슷한 값의 데이터가 빈번히 발생 시 하나의 센서 노드에 저장이 집중되는 문제가 있다. 기존의 데이터 중심 저장 방식에서 센서 데이터 저장 기법들은 저장 데이터의 효율적인 관리에만 치우쳐 센서 네트워크의 확장 시 하나의 센서 노드에 저장이 집중되는 문제점을 고려하지 않았다.

본 논문은 센서 네트워크의 확장 시 다차원 센서 데이터 저장의 효율적인 scalability를 지원하는 비균등 분할 기법을 제안한다. 제안한 기법은 센서 네트워크를 센서 노드의 분포에 따라 같은 센서 노드 개수를 갖는 영역으로 분할하고 분할된 각 영역 내에서 측정된 센서 데이터를 해당 영역에서 저장 및 관리함으로써 센서 네트워크의 확장에 따른 저장 비용을 줄였다. 그리고, 생성한 영역 개수를 센서 네트워크의 크기와 센서 노드의 개수, 발생하는 데이터의 양에 따라 증가시킴으로써 전체 센서 노드의 에너지 소모가 분산되어 기존의 방식과 비교하였을 때 센서 네트워크의 수명과 scalability가 향상되었다.

1. 서 론

센서 네트워크는 사람이 접근하기 힘든 산업 현장, 자연 현상 관측, 군사 시설 등 다양한 분야에서 사용되고 있다. 센서 네트워크는 일반적으로 적게는 수백에서 많게는 수백만 개의 센서 노드들로 구성되어 하나의 센서 노드는 여러 개의 센서들을 가지고 있으며 온도, 습도, 조도 등과 같은 다양한 스칼라 데이터를 측정한다[1,2].

본 논문은 데이터 중심 저장 방식(DCS: Data-Centric storage)[5]의 센서 네트워크를 기반으로 한다. 데이터 중심 저장 방식은 측정된 데이터 값에 따라 특정 데이터 키 값으로 분류되어 특정 센서 노드에 저장된다. 이 방식은 데이터가 많이 발생했을 때 외부 저장 공간에 가까운 센서들에게 부하가 집중되거나, 질의를 수행할 때 필요치 않은 센서가 질의 처리에 관여되는 것을 방지할 수 있다.

이에 센서 네트워크의 크기가 확장되어도 모든 센서 노드의 에너지 소비를 균등하게 하여 전체 네트워크의 수명을 연장하고 다차원 센서 데이터에 대한 영역 질의가 가능한 방법에 대해 활발히 연구되고 있다[3,4]. 그러나, 기존의 데이터 중심 저장 방식에서 다차원 센서 데이터 저장 기법들은 저장 데이터의 효율적인 관리에만 치우쳐 센서 네트워크의 확장에 따른 에너지 효율성을 고려하지 않았다.

따라서, 본 논문은 센서 네트워크를 센서 노드의 분포에 따라 같은 센서 노드 개수를 같도록 서로 다른 크기의 영역으로 분할하고, 분할된 각 영역 내에서 측정된 센서 데이터를 해당 영역에서 저장 및 관리함으로써 센서 네트워크의 확장에 따른 저장 비용과 특정 센서 노드에 통신이 집중되는 현상을 해결하는 방법을 제안하였다. 또한, 분할 영역 개수를 네트워크의 크기와 센서 노드의 개수, 각 센서 노드가 발생하는 데이터의 양에 비례하게 증가시켜 전체 센서 노드의 에너지 소모를 균등하게 하여 센서 네트워크의 수명을 연장하였다.

2. 관련 연구

2.1 DCS

DCS는 측정된 데이터를 값에 따라 특정 데이터 키 값으로 분류하고 데이터 키 값에 따라 해당 센서 노드에 저장하는 방식이다[5]. DCS는 데이터가 많이 발생했을 때 외부 저장 공간에 가까운 센서들에게 부하가 집중되거나 질의를 수행할 때 필요치 않은 센서가 질의 처리에 관여되는 것을 방지할 수 있다. 따라서, DCS는 센서 네트워크의 크기가 크고 측정 데이터의 저장이 많이 발생하는 환경에서 효율적이다.

2.2 GHT(Geographic Hash Table)

GHT는 DCS 기반 센서 네트워크에서 데이터의 값을 기반으로 지리적 위치를 생성하여 생성된 지리적 위치와 가장 가까운 센서 노드에 데이터를 저장하는 색인 기법이다[1]. GHT는 데이터 전송을 위해 GPSR 라우팅 기법을 사용한다[6]. GPSR 라우팅 기법은 데이터를 수신한 이웃 노드의 위치와 데이터의 목적지 정보를 사용하여 점진적으로 데이터 forwarding을 결정하는 Greedy Forwarding 기법과 경로를 우회하기 위한 Right Hand Rule 기법을 사용한다.

GHT는 센서 네트워크의 확장에서 센서 노드의 에너지 효율성을 위해 전체 센서 네트워크의 영역을 $4^d(d \geq 0)$ 개로 정분할하는 SR(Structured Replication) 기법을 사용한다. 그러나 SR 기법은 각 replication에 대표 센서 노드를 두기 때문에 hotspot의 통신비용이 높아진다.

2.3 DIM(Distributed Index for Multi-dimensional data)

DIM은 DCS 기반 센서 네트워크에서 범위 질의를 효율적으로 처리하기 위해 유사한 값을 가진 데이터가 동일한 혹은 지리적으로 인접한 센서 노드에 GPSR 라우팅 기법을 사용하여 저장시키는 인덱스 기법이다[2].

DIM은 비슷한 값을 갖는 데이터가 빈번히 발생되면 특정 센서 노드에 통신이 집중되어 센서 노드의 수명을 단축시키는 문제가 발생한다. 또한, 센서 노드의 추가로 인한 센서 네트워크

† 본 연구는 정보통신부 및 정보통신연구진흥원의 대학 IT 연구 센터 육성·지원사업의 연구결과로 수행되었음.

의 확장 시 데이터를 발생한 노드와 데이터를 저장할 노드와의 거리가 멀어지기 때문에 데이터 저장 비용이 높아지게 된다.

3. 비균등 영역 분할(NRS : Non-equal Region Split)

비균등 영역 분할 기법은 전체 센서 네트워크를 센서 노드의 분포에 따라 같은 센서 노드 개수를 갖도록 서로 다른 크기의 Region으로 분할하고, 분할된 각 Region 내에서 측정된 센서 데이터를 해당 Region에서 저장 및 관리하는 기법이다.

3.1 Region 생성

Region은 전체 네트워크 영역(x-y 평면)을 사각형 ROI라 할 때, 아래의 조건1을 만족하면서 R을 k번 분할하여 얻어진 사각형을 말한다.

조건 1: R은 i번 분할($1 \leq i \leq k$)하여 2^i 개의 사각형으로 나누어지며, 나누어진 각 영역의 센서 노드 개수는 동일하고, 나누어진 두 영역은 i가 홀수(짝수)면 분할은 y축(x축)과 평행함.

센서 노드의 분포가 균등하지 않은 상태에서 Region은 모두 같은 센서 노드 개수를 갖도록 분할되기 때문에 각 Region의 크기는 서로 다르며, Region 발생 개수는 2^i ($i>0$)이 된다. Region 코드는 분할 시 좌측 또는 하단 Region은 0, 우측 또는 상단 Region은 1을 부여한다. 생성할 Region 개수가 2이상이면 x, y축 분할을 반복하며 각 Region에 Region 코드를 부여한다.

Region 생성 알고리즘은 그림 1과 같다. 여기서 A는 전체 센서 네트워크 영역, Ra는 생성할 Region 개수, Rn은 생성된 Region 개수를 나타내며 RegionCode()는 입력 값을 기준으로 0 또는 1의 Region 코드를 할당하는 메서드이다.

Algorithm : Build-Region(A)

```

1: Begin
2:   while(Ra ≠ Rn)
3:     if(CountSensor(A) == Half_X) then
4:       Half_x = a.x;
5:       RegionCode(a.x);
6:     end if
7:     if(Ra == 2)
8:       break;
9:     if(CountSensor(A) == Half_Y) then
10:      Half_x = b.y;
11:      RegionCode(b.y);
12:    end if
13:   end while
14: End
    
```

그림 1. Region 생성 알고리즘

생성된 각 Region은 다차원 범위 질의 색인 기법인 DIM을 적용하여 Zone을 생성한다. 여기서 Zone은 영역(x-y 평면)을 사각형 ROI라 할 때, 아래 조건 2를 만족하면서 R을 k번 분할하여 얻어진 사각형을 말한다.

조건 2: R은 i번 분할($1 \leq i \leq k$)하여 2^i 개의 사각형으로 나누어지며 나누어진 두 영역은 i가 홀수(짝수)면 y축(x축)과 평행함.

조건 2에 의해 생성된 Zone은 서로 다른 Zone과의 구별을 위한 Zone 코드를 갖는다. Zone 코드는 x축 또는 y축으로 분할시마다 추가되는 비트로 구성되며, Zone이 좌측 또는 하단일 경우에는 0을 할당하고 우측 또는 상단일 경우에는 1을 부여한다.

3.2 데이터 저장

비균등 영역 분할 후 측정된 데이터의 저장 알고리즘은 그림 2와 같다. 여기서 Dv는 측정된 데이터 값, Zone_code는 Dv를 해싱하여 생성된 목적지 Zone code, GPSR_Route()는 데이터 발생 센서 노드의 이웃 노드들 중 목적 Zone에 가장 가까운 노드를 찾는 메서드이고 Except_neighbor()은 이웃하는 센서 노드 중 서로 다른 Region에 속한 이웃 센서 노드를 구별하는 메서드이다. 따라서 데이터의 저장은 최초 측정된 데이터 값을 해싱하여 저장될 위치의 Zone 코드를 생성하고 해당 Zone 코드에 위치한 센서 노드에게 GPSR 라우팅 기법을 사용하여 데이터를 전달하고 저장한다. 데이터의 저장은 동일 Region내에서 처리된다.

Algorithm : Storage-Data(Dv)

```

1: Begin
2:   Zone_code = Hash(Dv);
3:   while(Closed_code ≠ Zone_code)
4:     Closed_node = GPSR_Route(neighbor_node);
5:     if(Closed_node.Region != Zone_code.Region)
6:       Except_neighbor();
7:     end if
8:   end while
9: End
    
```

그림 2. 데이터 저장 알고리즘

그림 3은 센서 네트워크가 4개의 Region으로 분할된 예이다. 실선으로 그려진 사각형은 Region을 나타내고, 점선으로 그려진 사각형은 Zone을 나타낸다. 각 Region안의 00, 01, 10, 11은 Region 코드이다. 또한 각 Region에 포함된 센서 노드 개수는 같다. 그림 3과 같이 센서 네트워크의 영역을 분할하여 Region을 생성하고 측정된 데이터를 해당 Region의 센서 노드에 분산하여 저장함으로써 센서 노드의 확장 시에도 센서 데이터의 저장을 위한 통신비용과 Hotspot 통신비용을 줄일 수 있다. 여기서 Hotspot 통신비용은 비슷한 값을 갖는 데이터가 빈번히 발생 시 특정 센서 노드의 통신비용을 말한다.

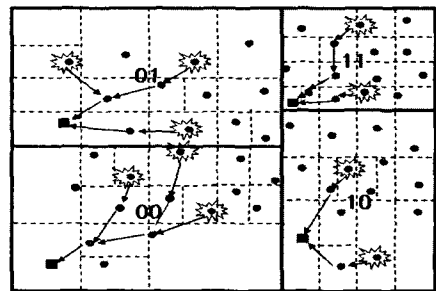


그림 3. Region 분할 시 센서 데이터 저장의 예

표 1은 전체 센서 네트워크의 영역을 N개의 Region으로 분할하였을 때 데이터 저장 시 발생하는 전체 저장 통신비용과 Hotspot을 보여준다. N은 Region의 개수, n은 센서 노드의 개수, $\sqrt{n_{total}}$ 은 센서 노드 n간의 point-to-point 라우팅 시 발생하는 통신비용, E는 측정된 센서 데이터 개수, E_{total}은 전체 센서 네트워크에서 발생하는 동일 이벤트의 발생 개수이다. 센서 네트워크를 N개의 Region으로 분할시 전체 통신비용은 분할 전과 비교하여 최대 N배 만큼 통신비용이 감소하고 또한 Hotspot도 최대 N배 만큼 감소하는 것을 알 수 있다.

표 1. N개의 Region 분할시 저장 비용

	분할 전	분할 후
전체 통신비용	$E_{total} \sqrt{N_{total}}$	$E_{total} \sqrt{N_{total}} / N$
Hotspot	E_{total}	E_{total} / N

3.3 질의 처리

그림 4는 질의 처리를 위한 알고리즘이다. 여기서 Qv는 질의 한 데이터 값, Zone_code는 Qv를 해쉬하여 생성된 Zone 코드, Rn은 생성된 Region 개수, SensorNum_R은 Region 내의 센서 노드 개수, Access_Node는 서로 다른 Region 간의 통신 노드, Def_AccessNode()는 Access 노드를 정하기 위한 메서드이다. 질의가 발생한 센서 노드는 전체 Region에게 각각 질의를 전달한다. 질의 발생 시 각 Region과 통신할 Access 노드를 결정한다. Access 노드가 결정되면 GPSR 라우팅 기법을 사용하여 질의를 전달한다. 질의를 전달받은 각 Region의 Access 노드는 자신이 속한 Region 내에서 질의 발생 노드가 되어 질의에 대한 데이터를 해싱하여 Zone 코드를 생성한다. 질의에 대한 응답은 질의 경로의 역 경로로 응답하게 된다.

Algorithm : Query-Data(Qv)

```

1: Begin
2:   Def_AccessNode();
3:   while(i < Rn)
4:     for(k=0;k<SensorNum_in_Region;k++)
5:       if(Region[i]==Access_Node) then
6:         Hash_code = Hash_Zone(Qv);
7:         while(Closed_code != Zone_code
8:           Closed_node = GPSR_Route(neighbor_node);
9:           if(Closed_node.Region != Zone_code.Region)
10:            Except_neighbor();
11:           end if
12:         end while
13:       end if
14:     end for
15:   end while
16: End
    
```

그림 4. 질의 처리 알고리즘

4. 성능 평가

4.1 저장 시 통신비용

저장 시 발생하는 통신비용의 성능 평가는 센서 노드의 통신 범위를 50m로 설정하고 센서 노드 개수를 100개부터 600개 까지 100개씩 증가시키면서 수행하였다. 저장 시 통신비용은 그림 5와 같다. 그림 5와 같이 Region 개수가 2, 4, 8, 16일 때 각각 Region의 개수에 따라 최대 71%, 87%, 92%, 94%까지 통신비용이 감소하였다.

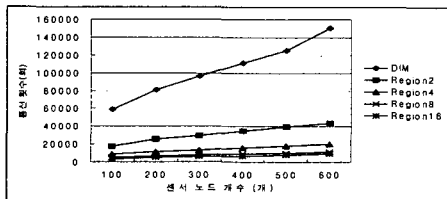


그림 5. 저장 통신비용 비교

4.2 Hotspot과 전체 통신비용

Hotspot 통신비용의 측정은 전체 센서 노드의 개수를 600개, 전체 네트워크의 크기는 600m x 600m, 센서 노드의 통신 범위는 50m, 데이터의 발생 범위는 전체 데이터 측정 범위의 10%이내에

서 랜덤으로 발생 하였다. Hotspot 통신비용은 그림 6의 (a)와 같이 Region 개수가 2, 4, 8, 16일 때 각각 Region의 개수에 따라 최대 66%, 68%, 56%, 41%까지 Hotspot이 감소하였다. 그리고 전체 통신비용의 성능 평가환경은 저장 통신비용의 성능평가 환경과 같으며 결과는 그림 6의 (b)와 같이 Region 개수가 2, 4일 때 각각 최대 44%, 24%까지 통신비용이 감소하였으나 Region 개수가 8, 16일 때 각각 최대 46%, 166%까지 증가하였다.

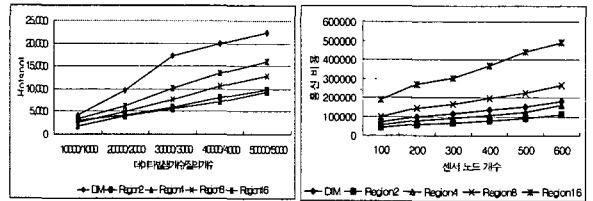


그림 6(a) Hotspot 비교 그림 6(b) 전체 통신비용 비교

5. 결론 및 향후 연구 과제

DCS 기반의 센서 네트워크에서는 센서 네트워크의 확장에 따른 데이터의 저장 비용의 증가와 특정 센서 노드의 통신 집중으로 인한 전체 네트워크의 수명에 상당한 영향을 주는 문제점이 있다.

본 논문에서 제안한 비균등 영역 분할 기법은 센서 네트워크를 센서 노드의 분포에 따라 같은 센서 노드 개수를 같도록 서로 다른 크기의 영역으로 분할하고 분할된 각 영역 내에서 측정된 센서 데이터를 해당 영역에서 저장 및 관리함으로써 센서 네트워크의 확장에 따른 저장 비용을 줄였다. 또한 분할 영역 개수를 네트워크의 크기와 센서 노드의 개수, 각 센서 노드가 발생하는 데이터의 양에 비례하게 증가시켜 전체 센서 노드의 에너지 소모를 균등하게 함으로써 센서 네트워크의 수명과 scalability를 향상시켰다.

향후 연구 과제로는 데이터 중심 저장 방식의 센서 네트워크 환경에서 비슷한 값의 데이터가 동일 센서 노드에 저장되기 때문에 특정 노드의 수명이 다하였을 경우 특정 범위의 데이터 손실이 발생한다. 이러한 데이터의 손실을 효율적으로 방지할 수 있는 방안이 필요하다.

참고 문헌

- [1] S. Ratnasamy, B. Karp, L. Yin, F. Yu, D. Estrin, R. Govindan, and S. Shenker, "Ght: A Geographic Hash Table for Data-centric Storage in Sensornets," Proc. on Wireless Sensor Networks and Applications, ACM, 2002.
- [2] X. Li, Y.-J. Kim, R. Govindan, and W. Hong, "Multi-dimensional Range Queries in Sensor Networks," Proc. on Embedded Networked Sensor Systems, ACM, 2003.
- [3] B. Greenstein, D. Estrin, R. Govindan, S. Ratnasamy, and S. Shenker, "Difs: A Distributed Index for Features in Sensor Networks," Proc. First IEEE International Workshop on Sensor Network Protocols and Applications, 2003.
- [4] V. Dyo, and C Mascolo, "Energy Efficient Distributed Index for Large Scale Sensor Networks," Proc. Workshop on Ubiquitous Computing and e-Research, 2005.
- [5] S. Ratnasamy, D. Estrin, R. Govindan, B. Karp, and S. Shenker, "Data-Centric Storage in Sensornets." SIGCOMM Computer Communication, ACM, 2002.
- [6] B. Karp, and H. T. Kung, "GPSR: Greedy Perimeter Stateless Routing for Wireless Networks," Proc. 6th Annual International Conference on Mobile Computing and Network, ACM, 2000.