공간 슬라이딩 윈도우 집계질의의 정확도 향상을 위한 그리드 해쉬 기반의 부하제한 기법

(Load Shedding Method based on Grid Hash to Improve Accuracy of Spatial Sliding Window Aggregate Queries)

백 성 하* 이 동 욱* 김 경 배** 정 원 일*** 배 해 영****

(Sung Ha Baek) (Dong Wook Lee)(Gyoung Bae Kim)(Weon il Chung) (Hae Young Bae)

요 약 데이터 스트림은 다양한 입력속도로 끊임없이 입력되고 데이터 스트림을 저장하는 메모리상의 저장공간은 유한하기 때문에 때때로 저장공간을 초과하는 데이터가 입력되는 경우가 발생한다. 이 문제를 해결하기 위해 초과되는 데이터의 일부를 버려 메모리 초과를 방지하는 부하제한 기법이 연구되었다. 기존의 부하제한 기법은 데이터의 편차에 따른 최적의 샘플링 비율을 갖는 랜덤 샘플링을 사용한다. 그러나 이 기법은 공간적 특성을 고려하지 않기 때문에 공간 질의에 사용되는 데이터와 사용되지 않는 데이터를 구분하지 않고 샘플링 한다. 그래서 공간 질의가 포함되는 u-GIS 환경에서는 질의 정확도가 감소하는 문제가 발생하였다. 본 논문에서는 공간 질의와 비공간 질의가 동시에 발생하는 u-GIS 환경에서 질의 정확도를 보다 향상 시키는 부하제한 기법을 연구하였다. 이 기법은 동시에 실행되는 공간 질의의 공간적 이용도에 따라 차등적으로 샘플링을 하여, 질의에 이용될 확률이 낮은 데이터를 샘플링을 한다. 제안된 부하제한 기법은 공간질의가 존재하는 경우 질의 정확도를 크게 향상 시켰고, 샘플링 중 공간 필터링 연산을 적용하여 질의 처리 속도도 일부 향상 시켰다.

키워드 : 스트림, 부하 제한, 공간 집계 질의, 윈도우 집계 질의, 연속 질의

Abstract As data stream is entered into system continuously and the memory space is limited, the data exceeding the memory size cannot be processed. In order to solve the problem, load shedding methods which drop a part of data to prevent exceeding the storage space have been researched. Generally, a traditional load shedding method uses random sampling with optimized rate according to data deviation. The method samples data not to distinguish those used in spatial query because the method uses only a random sampling with optimized rate according to data deviation. Therefore, the accuracy of query was reduced in u-GIS environment including spatial query. In this paper, we researched a new load shedding method improving accuracy of the query in u-GIS environment which runs spatial query and aspatial query simultaneously. The method uses a new sampling method that samples data having low probability used in query. Therefore proposed method improves spatial query accuracy and query processing speed as applying spatial filtering operation to sampling operator.

Keywords : Data Stream, Load Shedding, Spatial Aggregate Query, Window Aggregate Query, Continuous Query

† 본 연구는 건설교통부 첨단도시기술개발사업 - 지능형국토정보기술혁신 사업과제의 연구비지원(07국토정보C05)에 의해 수행되었습니다.

* 인하대학교 컴퓨터정보공학과 박사과정, {shbaek, dwlee}@dblab.inha..ac.kr

** 서원대학교 컴퓨터교육과 조교수, gbkim@seowon.ac.kr

*** 호서대학교 정보보호학과 전임강사, wnchung@hoseo.edu(교신저자)

**** 인하대학교 컴퓨터정보공학과 조교수, hybae@inha.ac.kr

논문접수 : 2008.12.29 수정일 : 1차 2009.01.31 심사완료 : 2009.02.02

1. 서 론

최근 통신 기술의 발달에 따라 다양한 센서가 등장하고 센서네트워크가 진보함에 따라 인간과 컴퓨터가 연결되는 유비쿼터스 환경이 도래되고 있다. 이 유비쿼터스 환경의 다양한 응용을 지원하기 위한 플랫폼 기술로 국토에 대한 공간 및 위치정보를 제공하는 u-GIS 공간정보 기술이 대두되고 있다[1,2]. 이 u-GIS 공간정보 기술은 기존 GIS인 건물, 도로, 하천, 지하시설물과 같은 2차원 또는 3차원상의 정적인 지형 지물 정보와 유비쿼터스 환경을 기반으로 시간에 따라 공간적인 위치가 포함된 동적인 GeoSensor 정보의 융합 처리를 요구한다.

GeoSensor는 고정된 지역이 아니라 넓은 지역에 산발적으로 분포될 수 있으며, 자신의 위치 인식 장치를 이용하여 시간에 따라 장소를 이동할 수 있는 이동성도 가지고 있다. 그래서 GeoSensor는 데이터 스트림의 특성과 GIS의 특성을 동시에 갖는다. GeoSensor는 넓은 지역에서 실시간으로 발생하는 대용량 정보를 처리하기 위해 데이터 스트림 처리 기술이 요구된다[3,4,5,18]. 또한 GIS 공간 정보와의 융합 처리를 위해 공간 연산 처리가 요구된다. 이와 같은 처리를 위해 일반적으로 공간 슬라이딩 윈도우 집계질의가 사용된다. 공간 슬라이딩 윈도우 집계 질의는 데이터 스트림 처리를 위한 연속질의에 집계 연산이 추가된 일반적인 슬라이딩 윈도우 집계 질의[3,4,5]의 프리디킷에 공간 연산이 포함된 질의이다. 예를 들면 “10초 동안 A 건물로부터 5km 내에 있는 사람들의 평균 연령을 5초마다 구하라”와 같이 슬라이딩 윈도우 질의와 공간 연산이 결합된 질의가 이에 해당된다.

그런데 이 GeoSensor에서 수집되는 데이터의 양은 실시간으로 변하고, 데이터를 저장하기 위한 공간은 한정적이기 때문에 이 유한한 공간을 초과하는 경우가 발생할 수도 있다. 일반적으로 데이터 스트림 처리기는 이 문제를 해결하기 위해 부하제한 기법을 사용 한다[6,7,8,9,10]. 특히 Brian의 연구에서는 일반적인 데이터 스트림 환경에서 집계 질의를 처리할 때 부하제한을 하기 위하여 정확도를 최대한 보장하는 샘플링 비율과 샘플링 속도를 최적화 하기 위해 부하제한 연산(Load Shedding Operator)의 위치를 선택하는 기법에 대해 논의하였다[11]. 이 기법에서 샘플링 비율은 집계 질의의 집계 연산의 대상이 되는 데이터들의 표준편차를 이용하여 편차가 클수록 데이터를 적게 샘플링하여 정확도 감소를 최소화하는 샘플링 비율을 선택하는 방법을 제시하였다. 또한 질의 스케줄에서 연산이 공유됨에 따라, 공유되는 연산과 질의가 가진 샘플링 비율을 가지고 중복 샘플링을 방지하기 위해 샘플링 비율을 분할하여 샘플링 연산을 배치하는 방법을 제시하였다.

그러나 이 방법은 공간 연산이 이용되는 u-GIS 환경에서는 비효율적인 면이 있다. 우선 이 방법은 샘플링을 할 때 랜덤 샘플링을 사용하게 된다. 랜덤 샘플링은 공간적 위치를 고려하지 않으므로, 특정 공간에 해당하는 데이터만이 질의 결과에 이용되는 공간 집계 연산의 경우 정확도를 감소 시킬 수 있다. 또한 샘플링 연산을 배치하는 기존 방법은 연산의 공유만 고려하여 비율을 분할하기 때문에 공간적 특성에 따라 차등적으로 비율을 분할할 수 없다.

이러한 문제점을 해결하기 위하여 본 논문에서는 공간적 특성을 반영한 부하제한 기법을 제안한다. 본 기법은 먼저 공간적 특성을 반영하기 위하여, 공간적으로 많이 이용되는 부분에 가중치를 둔 샘플링을 하는 부하제한 연산을 이용한다. 이와 같은 부하제한 연산을 빠르게 처리하기 위해서 데이터를 저장하는 스트림의 구조도 공간적으로 분할한 해쉬 구조를 이용한다. 또한 공간 연산이 포함된 질의 스케줄을 생성할 때, 공간 연산의 포함관계에 따라 질의 스케줄을 생성한다.

이와 같은 부하제한 기법은 u-GIS 환경에 적합한 공간적인 특성을 최대한 이용하게 되어 부하 제한 시 질의 정확도 및 처리 속도를 크게 향상 시킬 수 있다. 먼저 공간적으로 분할한 해쉬 구조를 사용하기 때문에 공간 연산의 처리 속도를 향상 시킬 수 있고, 공간적으로 데이터가 밀집된 곳에 가중치를 부여하여 공간적 이용도를 반영한 부하제한 연산을 지원할 수 있고, 질의 정확도 감소를 최소화할 수 있다. 또한 공간 연산이 포함된 질의 스케줄을 생성할 때, 공간 비공간을 동시에 고려한 스케줄을 사용하여, 공간 연산이나 비공간 연산의 샘플링을 통해 공간 적으로 편중된 샘플링을 하거나 공간적인 구분 없이 샘플링을 하는 문제를 방지할 수 있다.

2. 관련 연구

2.1 GIS와 데이터 스트림

유비쿼터스 시대를 위해 새롭게 요구되는 u-GIS 환경은 데이터 스트림 처리 시스템(DSMS: Data Stream Management System)과 지리정보 시스템(GIS : Geography Information System)이 결합된 플랫폼인 u-GIS DSMS를 요구한다. 지리정보 시스템에서 사용하는 공간 데이터를 지원하기 위해 현재 대부분의 상용 데이터베이스인 오라클, MySQL은 공간 데이터를 관계데이터베이스 상에서 지원한다[12,13]. 공간 데이터에 대한 표준화 및 스펙에 대한 정의는 OGC(Open GIS Consortium)에서 제공하고 있다[14,15].

u-GIS DSMS는 건물, 도로, 하천, 지하시설물과 같은 지형·지물들이 공간데이터 형식으로 기 구축된 공간 데이터베이스와 GeoSensor에서 발생되는 위치 정보가 포함된 실시간 데이터 스트림을 융합(조인)해서 유비쿼터스 환경의 다양한 응용지원을 하게 된다. 예를 들면 “최근 10초 동안 A업종의 건물 주변에 있는 사람들의 평균 나이를 5초마다 구하라”와 같은 공간 슬라이딩 윈도우 집계 질의의 처리를 필요로 한다. 따라서 u-GIS DSMS는 기존의 DSMS 연산과 공간 연산, 그리고 SDBMS로부터의 로딩 및 조인 기술을 필요로 한다.

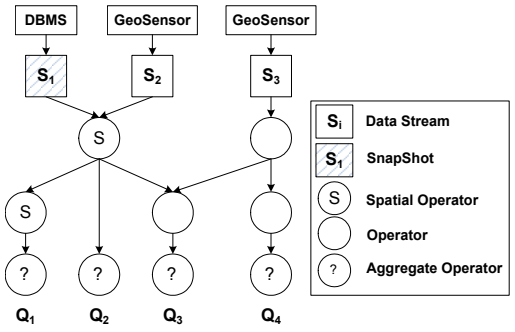

그림 1. u-GIS 환경에서 데이터 흐름 다이어그램

[그림 1]은 u-GIS DSMS에서 사용되는 데이터 흐름 다이어그램이다. S1과 같은 스냅샷(Snapshot)은 데이터베이스 내에 저장된 데이터의 일부를 메모리에 로딩하는 것으로 고정된 양의 데이터를 갖는다. S2와 같은 데이터 스트림(Data Stream)은 GeoSensor로부터 끊임없이 획득되는 데이터로 그 양이 시간에 따라 계속 변한다. 공간 연산(Spatial Operator)은 OGC 표준인 Contain, Overlap과 같이 공간객체간의 관계에 대한 연산들이다. 연산(Operator)은 일반 DBMS에서 제공하는 셀렉션, 프로젝션과 같은 연산이다. 집계 연산(Aggregate Operator)은 합, 평균, 카운트와 같은 연산이다. 이와 같이 공간 연산 및 스냅샷 데이터와의 공간 조인이 추가된 새로운 질의계획을 위해 공간적 특성을 고려한 부하제한 기법의 연구가 필요하다.

2.2 부하제한 기법

데이터 스트림의 입력속도가 실시간으로 변화하고 유한한 메모리를 사용하는 특징 때문에, 데이터 스트림의 저장공간이 초과되는 경우가 발생할 수 있다. 이와 같은 경우 DSMS는 스트림에 저장된 데이터의 일부를 버리는 부하제한 기법을 사용한다. 부하제한 기법은 데이터를 버리기 때문에 정확도를 감소 시킬 수 있다. 어떤 기준에 의해 언제, 얼마만큼의 데이터를 버리는지에 따라 정확도가 크게 달라질 수 있다.

질의의 정확도를 최대한 보장하고, 에러율을 최대한 균등하게 분산시키고 빠른 속도로 부하제한을 하기 위해, Brian은 질의 별 최적의 샘플링 비율과 질의계획에 따른 부하제한 연산의 위치를 결정하는 방법에 대해서 연구하였다[11]. 이 연구에서 부하제한 연산은 연산마다 샘플링 비율을 가지고 질의 수행 계획에 있는 일부 연산에 적용된다. 이 부하제한 연산은 랜덤 샘플링을 사용하여 주어진 샘플링 비율만큼 데이터를 감소시킨다. 만약 샘플링 비율(Pi) 값이 90이면 10%의 데이터를 삭제하는 것이다. 질의 정확도를 최대한 보장하는 샘플링 비율을 찾기 위해, Brian의 기법은 다음과 같은 방법을 사용한다.

$$\sum_{i \le i \le k} t_i r_{src(i)} p_i \prod_{O_x \in U_i} s_x p_x \le 1 \qquad \text{(식1)}$$

(식1)은 부하 방정식(Load Equation)이다. 이 식은 등록된 k개의 질의에 포함된 연산들의 합이 1을 넘으면 안 되다는 것으로, 데이터 입력 속도에 따른 샘플링 비율을 결정하는데 이용된다. t_i는 각 튜플 당 평균 처리 속도이고 $r_{src(i)}$는 데이터 스트림 입력율이고 p_i는 등록된 부하제한 연산의 샘플링 비율이다. $\prod$는 연산들의 선택율(Selectivity) s_x와 샘플링 비율 p_x를 포함한다. 그러므로 입력속도인 $r_{src(i)}$가 증가하면 p_i를 감소시켜야 한다.

$$C_i \ge \sqrt{\frac{\sigma_i^2 + \mu_i^2}{2N_i\mu_i^2} \log\frac{2}{\delta}}, \quad P_i \ge C_i / \varepsilon_i \qquad \text{(식2)}$$

(식1)을 만족하는 최적의 샘플링 비율을 구하기 위해 (식2)를 사용하게 된다. ε_i는 에러율 이다. 샘플링 비율 P_i는 C_i와 ε_i에 의해 결정된다. 그러므로 C_i가 증가하면 샘플링 비율이 증가하게 되는데, 데이터의 분포도를 나타내는 표준편차인 σ_i가 증가하면 C_i가 증가하여 샘플링 비율이 증가하게 된다. 또한 이 기법은 총 프로세싱 시간을 감소시키기 위하여 부하제한 연산의 위치를 결정하는 방법을 제안하였다.

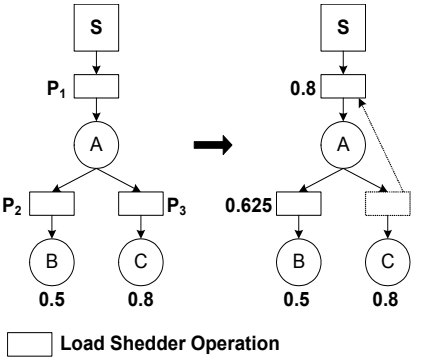

그림 2. 부하제한 연산 위치 결정 예

[그림 2]는 부하제한 위치를 결정하는 방법에 대한 예시이다. 두 개의 질의가 A라는 연산을 공유하고 첫 번째 질의의 샘플링 비율은 0.5 두 번째 질의의 샘플링 비율은 0.8이다. 이때 가장 높은 샘플링 비율인 C의 비율(0.8)을 갖는 부하제한 연산을 공유되는 연산으로 이동시키고, 이동 시킨 C의 비율로 B의 비율인 0.5를 나눈 값인 0.625 = 0.5/0.8을 샘플링 비율로 가지는 부하제한 연산을 A에 연결된 B의 질의의 다음 연산에 위치시킨다. 이와 같은 방법으로 연산 수행 전에 최대한 많은 샘플링을 수행하여 프로세싱 시간을 감소시킬 수 있다.

그러나 이 기법은 공간적인 특성을 고려하지 않고, 입력속도와 데이터 분산에 의존한 샘플링 비율을 사용하여, 오직 샘플링 비율에만 의존한 부하제한 연산의 위치를 선택하기 때문에 공간 데이터가 존재하는 경우 정확도와 처리 속도가 감소할 수 있다.

3. 적응적 메모리 관리 기법

본 논문은 u-GIS 환경에서 데이터의 공간적 특성을 반영하여, 공간적으로 고르게 부하제한을 하고 공간적 중요도에 따라 차등적으로 부하제한을 하여 질의 정확도와 처리 속도를 향상시키는 부하제한 기법인 UGLD(u-GIS Load Shedding)을 제안한다. UGLD는 데이터의 공간적 분포 및 중요도를 선정하기 위해 데이터 스트림을 공간적으로 분할하여 저장하는 데이터 스트림 관리 자료 구조를 사용한다. 이 자료 구조는 각 공간 영역별로 저장된 튜플 수와 데이터의 중요도를 가지고 있다. 이를 바탕으로 UGLD는 랜덤 샘플링을 사용하지 않고 공간 중요도별 가중치 샘플링을 사용한다. 그리고 공간 연산과 일반 연산이 같이 존재하는 질의 스케줄에서 공간 연산 공유 및 위치에 따른 부하제한 연산의 위치 및 샘플링 비율을 결정하는 방법을 사용한다.

그래서 공간적으로 많이 이용되는 부분에 가중치를 두어 상대적으로 샘플링을 적게 하여 질의 정확도를 향상시킬 수 있고, 데이터가 많이 입력된 영역을 좀 더 많이 샘플링 하여 데이터 양에 따라 샘플링을 균등하게 하도록 한다. 또한 연산이 공유되는 경우, 공간 샘플링을 우선 적용하게 되어 특정 공간 위주로 샘플링을 하면 공간 연산을 사용하지 않는 질의는 편중된 데이터에 영향을 받아 질의 정확도가 감소될 수 있기 때문에 공간, 비공간 연산을 모두 고려하여 부하제한 연산자의 위치를 결정하여 정확도 감소를 방지할 수 있다.

3.1 스트림 관리 공간 자료 구조

본 절에서는 UGLD에서 다루는 위치 정보를 포함한 GeoSensor 데이터 스트림을 공간적 특성에 따라 저장 관리하는 자료 구조인 그리드 기반 해쉬 스트림 큐(GHSQ: Grid Hash Stream Queue)에 대해서 설명한다.

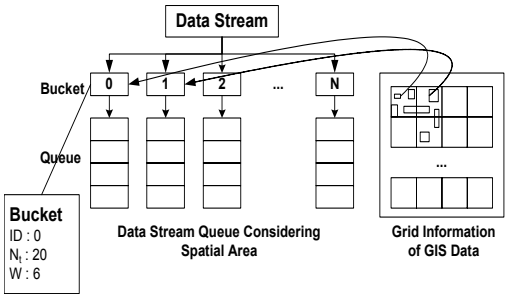

그림 3. 공간 연산 지원을 위한 그리드 기반 해쉬 스트림 큐

GeoSensor 데이터 스트림은 위치 정보를 포함하고 있기 때문에 주로 GIS 정보를 가지고 있는 공간객체와 조인되어 융합처리 되거나 공간 연산이 포함된 질의로 처리된다. [그림3]은 이 데이터 스트림과 조인되는 공간 객체 및 공간 연산의 영역이 주어지면, 이 공간 영역을 모두 포함하는 큰 영역을 만들고 이를 그리드 셀로 분할한다. 그리드의 분할 개수는 사용자가 임의로 결정할 수 있다. 이 그리드의 분할 개수만큼 데이터 스트림을 저장할 큐를 생성하고, 생성된 각 큐는 이 그리드 영역에 포함되는 데이터 스트림을 저장한다. 각 그리드 영역에 대응되는 큐를 가리키는 버켓(Bucket)은 큐의 정보와 큐와 연결된 그리드의 정보를 갖는다. 버켓이 저장되는 정보는 ID, Nti, Wi의 3가지로 ID는 버켓의 고유한 ID이고, Nti는 i 번째 큐에 저장된 튜플의 개수이다. 그리고 W는 가중치로 큐의 우선순위를 나타내기 위한 값으로, 버켓에 연결된 그리드 셀에 포함된 공간객체의 수에 비례하여 증가한다. 즉 W가 높을수록 해당 그리드가 많은 공간 객체 및 공간 연산의 영역을 포함하고 있으므로, W가 높을수록 질의 결과에 영향을 미칠 확률이 높은 데이터이다. 각 그리드 셀의 W는 해당 셀에 포함된 공간 객체의 수가 특정 단위를 초과할 때마다 증가하게 된다.

이 W_i는 최대 k이하의 자연수이고 기본적인 증가 단위로 L을 사용한다. k와 L은 사용자가 임의로 결정할 수 있다. 또한 공간 객체가 각 그리드 셀에 포함되거나 겹친 횟수를 GN_i 라고 하고 그 중 가장 큰 값을 M이라 한다. W_i의 결정은 기본적으로 설정된 L을 사용하지만,M 값에 따라 L의 값을 변경해야 하는 경우가 발생한다..

$$L = \left\lceil \frac{M}{k} \right\rceil, \; if \; kL < M \qquad (식3)$$

$$W_i = \left\lceil \frac{GN_i}{L} \right\rceil, M = Max(GN_i) \qquad (식4)$$

(식3)은 가중치의 증가 단위인 L을 변경하는 경우의 식이다. 위 식과 같이 M이 kL보다 크게 되면, 기본 L단위로 W_i를 계산하게 되어 최대 W_i의 값인 k를 초과할 수 있다. 그러므로 이 경우는 L의 범위를 M/k로 변경 한다. (식4)는 W를 구하는 식이다. 각 그리드 셀의 GN_i에 L을 나눈 값의 올림으로 W_i를 구한다.

이와 같은 방식으로 가중치를 계산하는 방법을 예를 통해서 설명한다. 먼저 4개의 그리드 셀이 주어지고 k를 4, L이 20이라고 가정하자. 첫 번째 그리드에 포함되거나 겹치는 객체의 수는 10, 두 번째 그리드는 32, 세 번째 그리드는 18, 그리고 마지막 그리드는 0이라고 하자. 그럼 M은 M = MAX(10,32,18,0) = 32이고 32는 M=32 < kL=80 이므로 기본 L 값을 사용할 수 있다. 10개의 데이터를 갖는 첫 번째 큐의 가중치는 $\lceil 20 \rceil$으로 1이다. 32는2, 18은 1, 마지막 0은 0이다. 이와 같은 방법으로 GHSQ을 구할 수 있다. 다음 절에서는 GHSQ를 가지고 공간적 특성에 기반한 샘플링을 하는 방법을 설명한다.

3.2 가중치 및 공간 밀집도를 이용한 공간 샘플링

본 절에서는 구축된 GHSQ을 가지고 공간 샘플링을 하는 방법에 대해서 설명한다. GHSQ는 각 버켓마다 연결

된 큐의 가중치와 큐에 저장된 튜플의 개수를 가지고 있다. 이 정보를 이용하여 UGLD는 그리드 셀 별로 부여된 가중치와 큐에 저장된 튜플의 개수를 기반으로 공간 샘플링을 한다. 가중치는 해당 셀에 있는 데이터가 공간 조인이나 공간 연산에 상대적으로 많이 이용될 수 있음을 나타내는 수치이다. 큐에 저장된 튜플의 개수가 많을수록 상대적으로 한 개의 튜플이 질의 결과에 미치는 영향이 작은 것이다. 예를 들면 10개의 튜플에서 1개의 튜플이 그 튜플들의 평균에 미치는 영향이 100개의 튜플에서 1개의 튜플이 미치는 영향보다 일반적으로 크다. 공간 조인은 일반적으로 공간 연산을 가지고 특정 영역별로 집계를 하고 결과를 반환하는 질의이다. 그래서 공간적으로 비슷한 위치에 있는 여러 데이터가 한 개의 집계 값의 계산에 이용되므로, 비슷한 위치인 같은 그리드 셀에 입력되는 데이터들이 특정 집계 값을 계산하는데 이용될 가능성이 높다.

이와 같은 특징을 이용하여, 각 그리드 셀이 가지고 있는 가중치에 반비례하고, 튜플 수에 비례한 샘플링 비율을 결정하는 방법을 설명한다. 각 질의는 질의마다 기본적인 샘플링 비율을 갖는다고 가정한다. 이 비율은 Brian의 방법과 동일하게 데이터의 표준편차와 입력속도에 따라 결정된다. 질의의 주어진 샘플링 비율이 Pi라고 하면, 이 비율을 그리드 셀 별로 분배해야 한다. 이 분배 규칙은 각 셀의 가중치와 튜플 수를 가지고 결정한다.

일반적으로 튜플 수만 가지고 샘플링 비율을 적절하게 분배하기 위해서는 각 셀 별로 Pi의 비율만큼 샘플링을 해주면 전체적으로 Pi만큼 샘플링이 되면서 셀 별로 동일한 비율로 분배가 된다. 여기에 셀 별 가중치를 고려하면 셀 별로 적용되는 비율이 약간 달라진다. 셀 별 적용되는 샘플링 비율을 P_k^w 라 할 때, 셀 별 적용되는 비율을 변경해도 샘플링 양을 같게 하기 위해서는 다음 (식5)을 만족해야 한다.

$$\sum_{k=1}^{N}(1-P_k^W)N_k^t = (1-P_i)S \qquad \text{(식5)}$$

(식5)를 만족하면서 가중치 W가 높은 그리드 셀의 샘플링 비율을 낮추기 위해 다음과 같이 각 셀 별로 가중치가 적용된 튜플 수 인 N_k^w 을 사용한다.

$$N_k^w = 0, \text{if } W_k = 0$$
$$N_k^w = N_k^t \times (1-\alpha W_k), otherwise \qquad \text{(식6)}$$

(식 6)은 가중치가 적용된 튜플 수인 N_k^w 을 결정하는 방법이다. 가중치가 0인 경우는 공간 연산에 이용되지 않으므로 N_k^w 을 0으로 한다. 0이 아닌 경우에는 $(1-\alpha W_k)$ 을 튜플 수에 곱하는데 α 는 가중치별로 샘플링 비율을 정하는 값으로 α 값이 높아질수록 가중치가 샘플링을 비율에 미치는 영향이 크게 된다. 이와 같이 가중치를 적용한 튜플 수가 주어지면 (정리1)에서와 같이 가중치가 적용된 샘플링 비율을 얻을 수 있다.

[정리 1]

$S = \sum_{k=1}^{N} N_k^t, S_w = \sum_{k=1}^{N} N_k^W$ 라 할 때 $P_k^W = 1 - \frac{N_k^W(1-P_i)S}{S_W N_k^t}$ 을 셀별로 적용되는 가중치 샘플링 비율로 사용하면, (식5)가 만족된다.

Proof) 증명은 (식5)에 $P_k^W = 1 - \frac{N_k^W(1-P_i)S}{S_W N_k^t}$ 을 대입하여 풀면 간단히 증명된다.

[정리 1]은 가중치를 적용한 샘플링 비율을 위와 같이 사용하여도 총 샘플링 되는 양이 같다는 것을 보인다. 그러나 여기서 고려하지 않은 것이 있다. 가중치가 0인 경우 P_k^w 는 1이 되기 때문에 이 경우는 샘플링을 전혀 하지 않는다. 가중치가 0인 경우는 공간 연산에 전혀 사용되지 않는 것이므로, 가중치가 0인 경우는 데이터를 모두 버릴 수 있는 것이다. 가중치가 0인 경우의 셀의 집합을 *Wo* 라 하면, 가중치가 0인 셀의 큐에 저장된 모든 데이터를 버리면 부하 제한을 통해 버리는 총 데이터의 양은 다음과 같다.

$$(1-P_i)S + \sum_{k \in Wo} N_k^t \qquad \text{(식7)}$$

사실상, $\sum_{k \in Wo} N_k^t$ 은 공간 연산에 이용되지 않는 데이터를 미리 필터링하는 것과 같으므로, 부하제한이 불필요한 상황에도 적용될 수 있다. 이와 같은 사전 필터링은 부하제한 발생 이전에도 데이터를 버리기 때문에, 부하제한 발생을 최소화할 수 있는 장점이 있다. 그리고 부하제한 발생 시 데이터 이용도에 따라 필터링을 하기 때문에 질의 정확도 향상에 크게 기여할 수 있다.

3.3 부하제한 연산의 위치 선정

지금까지 공간적 특성을 이용하여 샘플링을 적용한 부하제한 연산에 대해 설명하였다. 이번 절에서는 연산이 공유되는 스케줄 상에서 처리 속도를 향상시키기 위한 부하제한의 위치를 결정하는 방법을 설명한다.

기존 방법은 공간 연산을 고려하지 않기 때문에 공유되는 연산에 두 질의의 샘플링 비율 중 큰 값을 샘플링 비율로 갖는 부하제한 연산을 적용하고, 샘플링 비율이 낮은 질의 쪽에 남은 샘플링 비율을 한 번 더 적용하여 연산 적용 전에 최대한 많은 샘플링을 하도록 하는 방법을 사용 하였다.(관련연구2 참고) 그런데 비공간 연산이 최 상단에서 공유가 되고, 공간 연산과 비공간 연산을 갖는 두 개의 질의가 존재 한다면 정확도를 감소시킬 수 있는 문제가 있다.

[그림 4]는 비공간 연산이 공유되는 상황에 공간 연산을 갖는 질의 Q2가 존재하는 경우이다. 부하제한 연산(Load Shedder Operator)은 가중치를 고려하지 않고 일반적인 랜덤 샘플링을 하는 연산이다. 공간 부하제한 연산(Spatial Load Shedder Operator)는 3.2절에서 제안한 가중치를 적용한 샘플링 연산이다. 위와 같은 상황에서, 공간 연산을 갖는 Q2의 질의가 샘플링 비율이 높으므로 우선 0.8비율로 샘플링을 하면 이 비율에 해당하는 공간

부하제한 연산을 적용할 것이다. 그런데 Q1은 공간 이용도와는 아무 연계가 없는 비공간 질의이다. 그래서 Q2질의를 위해 공간 샘플링을 적용하게 되면 특정 공간에 대한 데이터를 주로 획득하게 되어 질의 정확도가 감소하는 상황이 발생할 수 있다. 이런 상황은 비공간 연산이 공유되는 경우 발생한다.

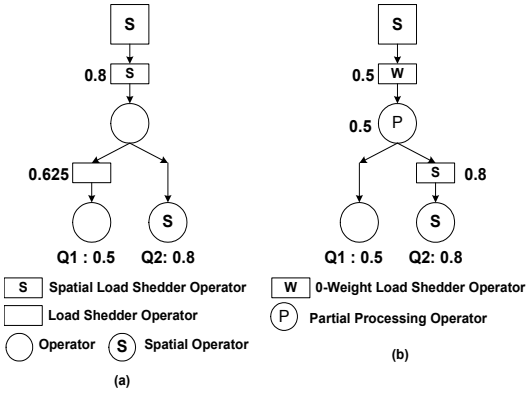

그림 4. (a) 공간 연산이 공유된 스케줄 상의 부하제한 (b) 공간-비공간 연산 공유를 위한 부분 부하제한 연산이 추가된 스케줄

표 1 연산 공유에 따른 정확도

번호	정확도	공유연산	연산 (high rate)	연산 (low rate)
1	O	공간	비공간	비공간
2	O	공간	비공간	공간
3	O	공간	공간	비공간
4	O	공간	공간	공간
5	O	비공간	비공간	비공간
6	X	비공간	비공간	공간
7	X	비공간	공간	비공간
8	X	비공간	공간	공간

표 1은 공유되는 연산에 따른 질의 정확도의 상실 여부에 대한 리스트이다. 6,7,8번의 경우 질의 정확도가 상실될 수 있다. 6번은 비공간 연산의 샘플링 비율이 높은 경우이다. 그래서 우선 비공간 연산의 샘플링 비율만큼 샘플링을 하면, 공간적으로 중요한 부분의 고려 없이 무조건 균등하게 샘플링을 하여 공간적으로 중요한 부분의 데이터를 상실할 수 있다. 7번의 경우는 [그림 4]와 동일하다. 8번의 경우는 어느 한쪽의 공간적 특성으로 샘플링을 하게 되면 다른 쪽 질의에 영향을 미칠 수 있다. 이런 문제를 해결하기 위하여 가중치를 기반으로 한 부분 부하제한 연산(WLSO : 0-Weight Load Shedder Operator)과 부분 처리 연산(PPO: Partial Processing Operator)을 제안한다.

가중치를 기반으로 한 부분 부하제한 연산은 비공간 연산이 공유되고 공간 비공간 연산이 각각 질의 상에 존재할 때 사용하는 것으로, 공간 연산에 영향을 주지 않는 가중치가 0인 데이터들을 우선 부하제한 하는 것이다. 0 이외의 가중치를 갖는 데이터는 연산과정에서 샘플링 되면서 처리 된다. 이를 부분 처리 연산이라고 한다.

[그림 5]는 비공간 연산이 공유되고 비공간 연산과 공간 연산을 갖는 질의의 스케줄이다. 이 스케줄에는 [그림 4]의 스케줄과 다르게 WLSO와 PPO연산이 추가되어 있다. WLSO는 가중치가 0인 셀의 큐에 저장된 데이터만 주어진 샘플링 비율로 버리고, 가중치가 0보다 큰 셀의 데이터는 PPO연산에 의해 선별적으로 처리된다. PPO연산은 비공간 질의를 위해서 샘플링된 데이터와 샘플링 되지 않은 가중치 1 이상의 데이터를 해당 질의의 샘플링 비율만큼 처리하여 비공간 질의가 요구하는 샘플링 비율만큼의 데이터를 처리한다. 또한 이 연산은 공간 질의를 위해 비공간 질의를 위해 처리되지 않은 남은 데이터를 처리하고 그 결과를 공간 부하제한 연산으로 전달한다.

WLSO연산과 PPO연산이 추가되는 실행계획은 다음과 같은 규칙으로 생성된다. WLSO연산은 [표1]의 6,7번과 같이 비공간 연산이 공유될 때, 비공간 질의의 샘플링 비율을 가지고 공유연산 위에 삽입된다. 공유된 연산은 WLSO와 동일한 샘플링 비율을 갖는 PPO 연산으로 변경된다. 그리고 PPO연산에 연결된 공간 연산의 위에 공간 질의의 샘플링 비율을 갖는 공간 부하연산이 삽입된다.

PPO 연산은 비공간 연산을 위하여, 공간적으로 균등하게 샘플링 비율을 적용하기 때문에 공간 부하제한으로 발생 가능한 정확도 상실 문제를 해결할 수 있고, WLSO 연산은 공간 연산에 이용되지 않는 부분만 샘플링 하고, PPO연산은 공간 질의에 의해 사용 가능한 모든 데이터를 처리하기 때문에 공간적으로 중요한 부분을 상실하지 않게 된다. 이 방법은 비공간 연산 공유에 따른 정확도 상실문제를 해결할 수 있다. 그러나 이 방법은 가중치가 0인 경우만 샘플링 하기 때문에 공유되는 연산이 처리해야 하는 데이터의 양이 증가할 수 있다. 증가량은 다음 식을 통해 확인 가능하다. R_i^a 는 GHSQ의 데이터 중 가중치가 0인 셀의 큐에 저장된 데이터의 비율이고 $R_i^s = 1 - R_i^a$ 는 가중치가 0이 아닌 데이터의 비율이다. 이 때 데이터 증가량 IN_i은 다음과 같다. S는 GHSQ에 저장된 데이터 총 수이다.

$$IN_i = (1-P_i)S - (1-P_i)R_i^a S = (1-R_i^a)(1-P_i)S \qquad (식8)$$

(식 8)은 R_i^a 가 증가하면 데이터 증가량 IN_i가 감소함을 나타낸다. R_i^a 의 비율이 1에 근사하게 되면 추가적인 처리를 요구하는 데이터가 거의 없게 된다. 즉 가중치가 0인 셀이 증가하면 데이터 처리 속도는 기존 기법과 거

의 동일하다. 일반적으로 공간 질의나 공간 조인의 경우 모든 영역에 해당하는 데이터를 요구하는 경우는 극히 드물다. 예를 들면 영역 질의(Range Query)나 근접 질의(k-nearest Query)와 같은 경우는 특정 위치나 객체의 주변을 검색하므로 가중치가 0인 셀이 많이 발생한다. 또한 조인 질의의 경우는 일반적으로 "서울시 내의 경찰청 주변에 있는…."과 같이 특정 공간객체를 추출한 후 조인하기 때문에 대부분의 셀이 가중치가 0이 넘는 상황은 극히 드물다. 물론 많은 공간 객체와 조인을 해야 하는 질의가 요청될 수도 있다. 이 경우는 가중치가 0이 넘는 셀이 많이 발생할 것이다. 그러나 처리해야 하는 데이터의 양이 증가하는 연산은 비교적 연산속도가 빠른 비공간 연산이고, (식8)과 같이 추가적으로 처리되는 양도 샘플링 될 비율만큼만 추가적으로 연산하기 때문에 계산량이 크게 증가하지 않는다. 그리고 GHSQ는 자체적으로 데이터를 공간 그리드 인덱싱을 하고, 부하제한 과정에서 데이터 사전 필터링을 수행하기 때문에 공간 연산의 처리 속도가 향상되어 추가연산으로 인한 큰 성능저하가 발생하기 어렵다. 또한 공유되는 연산에 적용되는 샘플링 비율은 질의의 샘플링 비율 중 가장 낮은 것이 선택되기 때문에 공유연산이 매우 큰 샘플링을 하는 경우는 일반적으로 드물다.

4. 성능분석

4.1 평가환경

실험 평가에 사용된 시스템 환경은 CPU가 펜티엄 4 3.0 GHz이고 메모리는 4GB이다. 시스템에서 할당하여 사용하는 메모리는 512MB이다. 실험은 이동성을 갖는 GeoSensor 데이터와 유사한 환경을 위해 이동체 데이터를 생성하는 프로그램인 IBM의 City Simulation을 이용한다[16]. City Simulation은 동시에 이백만 개의 이동객체를 생성할 수 있다. GIS데이터는 TIGER/Line 2007 데이터를 상용 DBMS인 오라클 형식으로 저장하여 실험에 이용한다[17]. 이 두 종류의 데이터를 이용하여 공간 데이터가 다양하게 존재하는 환경에서 다수의 공간, 비공간 질의를 이용하여 성능분석을 수행한다. 이동 객체를 저장하는 초기의 큐는 10MB의 크기로 할당하였다. 이동객체의 스키마는 <ID, Time, X, Y>이고 크기는 16바이트이다. 이동 객체는 초당 지정된 최대 크기까지 랜덤한 양으로 생성된다. 이와 같이 데이터를 생성하는 데이터 스트림은 총 10개를 사용한다.

4.2 성능평가

본 논문에서 제안하는 기법의 우수성을 증명하기 위해, 기존의 부하제한기법인 Brian의 방법과 제안된 기법인 UGLD를 비교한다. UGLD의 u-GIS 환경에서 공간적 이용도에 따라 샘플링을 하여 질의 처리 속도와 질의 정확도를 최대한 유지하는 특징은 다음 실험을 통해 확인한다. UGLD는 공간 연산을 공유하는 경우와 비공간 연산을 공유하는 경우가 차이가 있으므로 두 방법 모두 실험에 사용한다. 실험 평가 요소는 부하제한 빈도수와 질의 정확도, 비공간 연산 공유시의 질의처리 속도이다.

4.3 부하제한 빈도 수 측정

UGLD는 공간 질의가 실행되는 경우 공간적으로 이용되지 않는 부분에 대한 필터링으로 인해 부하제한 빈도를 크게 감소시킬 수 있다. 이를 확인하기 위해, 공간 질의와 비공간 질의가 동시에 실행되는 상황에서 동시에 발생하는 이동객체의 수를 증가시키면서 부하제한 연산이 발생하는 빈도수를 측정한다. 비교 방법은 우선 Brian의 방법과 비공간 질의를 공유하고 있는 UGLD(비공간), 마지막으로 공간 질의를 공유하고 있는 UGLD(공간) 3가지 방법을 비교한다. 실험 결과는 다음과 같다.

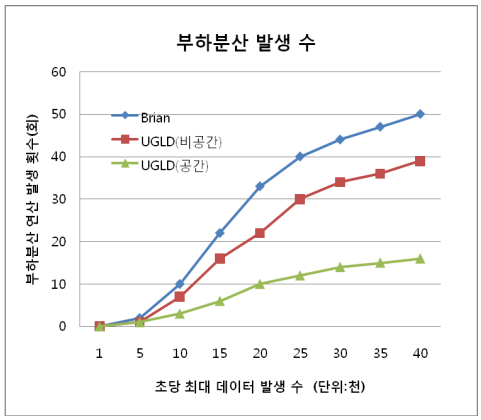

그림 5. 데이터 입력 증가에 따른 부하제한 연산

[그림 5]은 부하제한 연산 발생 수에 대한 분석 결과이다. Brian은 상대적으로 부하제한 연산이 많이 발생하고 UGLD(공간)이 가장 적게 발생한다. UGLD(비공간)은 Brain의 결과보다 적게 발생한다. 공간 연산을 공유하는 경우는 부하제한 전에 필터링 되는 양이 많기 때문에 부하제한의 발생을 사전에 크게 감소시킬 수 있다. UGLD(비공간)은 공간 연산을 공유하지 않고 일부 데이터를 추가적으로 처리하지만, 공유된 공간 연산의 사진 필터링을 통한 일부 처리 속도 향상 때문에 Brian과 비슷한 결과를 보인다.

4.4 질의 정확도 측정

다음 실험은 UGLD을 이용하는 경우의 질의 정확도를 분석하는 것이다. UGLD는 부하제한 발생률을 최소화하고 공간적 특성을 이용하여 정확도를 최대화하기 때문에 기존 기법보다 상대적으로 정확도가 우수하다. 특히 실행되는 공간질의가 많은 경우 정확도가 우수할 가능성이 높고, 공간 질의가 조인되거나 연산을 수행할 때 공간 질

의의 검색 범위가 작은 경우 우수할 가능성이 높다. 우선 공간 질의가 실행되는 환경에서 데이터 스트림의 입력속도 증가에 따른 질의 정확도를 측정한다. 이 경우도 4.2.1절과 동일하게 3가지 방법을 가지고 비교한다.

[그림 6]은 데이터 입력 증가에 따른 질의 정확도 비교이다. UGLD(공간)의 경우 부하제한 연산의 빈도가 가장 낮기 때문에, 가장 우수한 성능을 보인다. 다음은 공간 질의의 검색 범위의 크기에 따른 정확도의 측정이다. UGLD는 공간 조인이나 범위 연산이 있는 경우 해당 공간 연산이 처리를 요구하는 영역이 큰 경우 가중치가 0이 아닌 셀의 수가 증가하게 된다. 그러므로 공간 조인의 경우 조인하는 공간질의 결과(스냅샷)가 클수록 질의 정확도가 낮아질 가능성이 높다. 이를 측정하기 위해 공간 질의의 검색 비율에 따른 질의 정확도를 측정하였다.

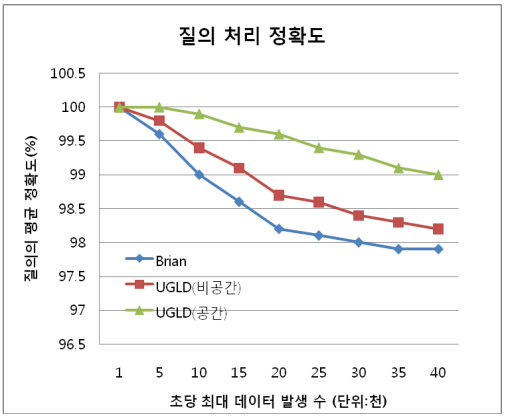

그림 6. 데이터 입력 증가에 따른 질의 정확도

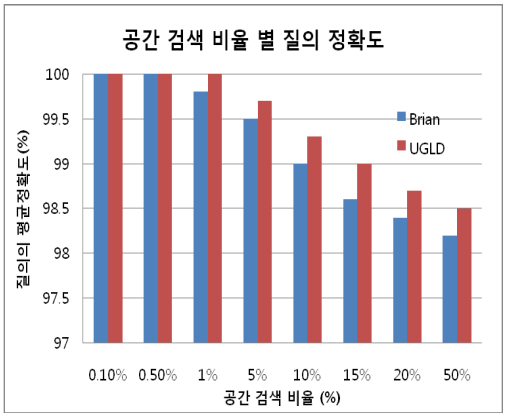

그림 7. 공간 질의의 검색 범위에 따른 질의 정확도

[그림 7]은 공간 질의의 검색 범위가 증가함에 따른 질의 정확도 분석 결과이다. UGLD는 1% 이하의 검색을 하는 경우 질의 정확도가 100% 보장된다. 이 경우는 부하제한이 발생하지 않기 때문에 가능하다. 그러나 공간 검색 비율이 증가하는 경우 UGLD의 정확도가 감소하는데 전반적으로 부하제한 발생 수가 크지 않고 공간 이용도에 따른 차등적 부하제한으로 인해 Brian보다 전반적으로 좋은 질의 정확도 결과를 보인다.

4.5 비공간 연산 공유 시 질의 처리 속도 비교

UGLD기법은 비공간 연산 공유 시 공간 연산에 사용되지 않는 일부 데이터를 공유된 비공간 연산에서 처리하는 상황이 발생할 수 있다. 이 경우 다소 불필요한 연산을 수행할 수 있지만 그 양이 크지 않고, 다음 공간 연산에서 속도 향상이 크기 때문에 전반적으로 성능저하에는 영향을 미치지 않는다. 이를 위해 비공간 질의 공유 시 질의처리 속도를 비교하였다. 비교 방법은 부하제한을 사용하지 않는 경우의 질의처리 속도를 100%로 하였을 때 Brian과 UGLD가 질의 처리에 걸리는 속도를 비율로 분석하였다.

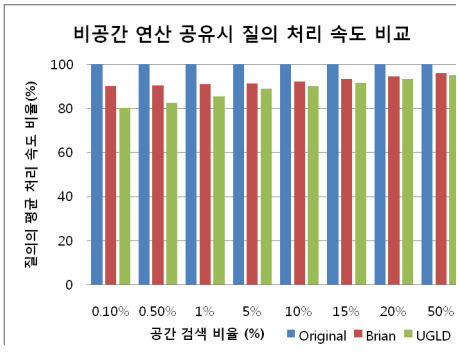

그림 8 부하제한 빈도수의 비교 분석

[그림 8]는 비공간 연산에 연결된 공간 연산의 공간검색 비율이 증가할 때 질의 처리 속도를 비교 분석한 결과이다. UGLD방법이 Brian에 비해 전반적으로 질의 처리 속도가 빠르다. 비록 비공간 연산 공유로 인해 일부 데이터에 대한 연산 부담은 증가하지만, 공간 연산 처리 시 감소하는 비용이 크게 증가하고 검색 비율이 작은 경우 큰 영향을 미치지 않는다.

5. 결 론

본 논문에서는 공간 질의가 존재하는 u-GIS 환경에서 효율적인 부하제한 기법을 제안하였다. 제안 기법은 공간 특성을 반영하여 공간적으로 이용되지 않는 부분은 필터링을 하고 이용되는 정도에 따라 차등적으로 샘플링을 하였다. 그래서 공간 질의가 사용되는 경우, 질의 정확도가 크게 향상되었다. 또한 공간 연산과 비공간 연산이 공유되는 상황에서 정확도와 질의 속도를 최대한 보장할 수 있는 연산 스케줄 방법에 대해서 설명하였다.

향후 연구는 그리드 셀에 가중치만 이용하여 샘플링을

하지 않고, 셀 별 입력되는 데이터의 표준편차를 이용하여 샘플링 비율을 보다 동적으로 최적화 하는 방법이다. 이를 바탕으로 가중치 적용을 보다 최적화하는 방법의 연구가 가능하다.

참 고 문 헌

[1] 이충호, 안경환, 이문수, 김주완. u-GIS 공간정보 기술 동향.

[2] 안경환, 김주환, "모바일 u-GIS 데이터 처리 시스템 설계.", 한국정보처리학회 추계 학술발표대회, 2008.

[3] Babcock, B., Babu, S., Datar, M., Motwani, R. and Widom, J., "Models and Issues in Data Stream Systems." PODS, 2002.

[4] Abadi, D. J, Carney, D., "Aurora: A New Model and Architecture for Data Stream Management." VLDB Journal, 2003.

[5] Golab, L., Tamer Ozsu. M., "Issues in Data Stream Management"" SIGMOD Record. ACM. Vol.32 No. 2, 2003, pp. 6-14.

[6] Tatbul, N., Cetintemel, U., Zdonik. S., Cherniack, M., and Stonebraker. M., "Load shedding in a data stream manager." VLDB, 2003.

[7] Motwani, R., Widom, J., Arasu, A., "Query processing, approximation, and resource management in a data stream management system.", CIDR, 2003.

[8] Tatbul, N., and Zdonik. S., "Load Shedding in a Data Stream Manager.", VLDB, 2003.

[9] Chakrabarti, K., Garoflakis, M., Rastogi, R., "Approximate Query Processing Using Wavelets," VLDB, 2000.

[10] Reiss, F. and Hellerstein, J., "Data Triage: An Adaptive Architecture for Load Shedding in TelegraphCQ.", ICDE, 2005.

[11] Babcock, B., Datar, M., and Motwani, R., "Load Shedding for Aggregation Queries over Data Streams", ICDE, 2004.

[12] "Oracle Spatial User's Guide and Reference 10g Release 1 (10.1)", Part No. B10826-01, www.oracle.com, 2003.

[13] "MySQL 6.0 Reference Manual", www.mysql.com, 2008.

[14] "OpenGIS Implemenation Specification for Geographic information - Simple feature access - Part1:Common Architecture", www.opengeospatial.org, 2008

[15] "OpenGIS Implemenation Specification for Geographic information - Simple feature access - Part1:SQL Option", www.opengeospatial.org, 2008

[16] Kaufman, J., Myllymaki, J., and Jackson, J., "City Simulator." Alpha Works Emerging Technologies, Nov. 2001.

[17] "Tiger/Line Shapefiles.", www.census.gov/geo/www/tiger/tgrshp2007/tgrshp2007.html, 2007.

[18] 강홍구, 박치민, 홍동숙, 한기준, "공간 센서 데이터의 효율적인 실시간 처리를 위한 공간 DSMS의 개발", 한국공간시스템학회 논문지, 제9권제1호, 2007, pp. 45-57.

백 성 하

2005년 인하대학교 컴퓨터공학부 졸업 (이학사)
2007년 인하대학교 컴퓨터 정보공학과 (공학석사)
2007년~현재 인하대학교 컴퓨터 정보공학과(박사과정)
관심분야는 데이터 스트림, 클러스터, 위치기반 서비스

이 동 욱

2003년 상지대학교 전자계산공학과 (이학사)
2005년 인하대학교 컴퓨터 정보공학과 (공학석사)
2005년~현재 인하대학교 컴퓨터 정보공학과(박사과정)
관심분야는 공간데이터웨어하우스, 공간정보관리, 유비쿼터스 환경을 위한 SDBMS

김 경 배

1992년 인하대학교 전자계산공학과 (공학사)
1994년 인하대학교 전자계산공학과 (공학석사)
2000년 인하대학교 전자계산공학과 (공학박사)
2000년~2004년 한국전자통신연구원 (선임연구원)
2004년~현재 서원대학교 컴퓨터교육과 조교수
관심분야는 이동실시간 데이터베이스, 스토리지 시스템

정 원 일

1998년 인하대학교 전자계산공학과 (공학사)
2004년 인하대학교 컴퓨터 정보공학과 (공학박사)
2004년~2006년 한국전자통신연구원 (선임연구원)
2007년~현재 호서대학교 정보보호학과 (전임강사)
관심분야는 데이터베이스, 데이터스트림, 이동객체

배 해 영

1974년 인하대학교 응용물리학과
(공학사)
1978년 연세대학교 대학원 전자계산학과
(공학석사)
1989년 숭실대학교 대학원 전자계산학과
(공학박사)
1985년 Univ. of Houston 객원교수
1992년~1994년 인하대학교 전자계산소 소장
1982년~현재 인하대학교 컴퓨터공학부 교수
1999년~현재 지능형 GIS연구센터 센터장
2000년~현재 중국 중경우전대학교 대학원 명예교수
2004년~2006년 인하대학교 정보통신대학원 원장
2006년~2009년 인하대학교 대학원장
관심분야는 분산 데이터베이스, 공간 데이터베이스, 지리정보 시스템, 멀티미디어 데이터베이스