

# 공간 데이터스트림의 입력 빈도와 데이터 밀집도 기반의 동적 부하제한 기법

정원일\*

<sup>1</sup>호서대학교 정보보호학과

## Dynamic Load Shedding Scheme based on Input Rate of Spatial Data Stream and Data Density

Weonil Jeong<sup>1\*</sup>

<sup>1</sup>Dept. of Information Security Engineering, Hoseo University

**요약** u-GIS 환경에서는 실시간으로 유입되는 공간 데이터 스트림으로 인해 발생하는 부하를 제한하기 위한 연구가 계속되고 있다. 그러나 기존의 비공간 데이터 기반의 부하 제한 기법은 공간 데이터의 특성을 고려하지 않아 공간 질의 처리의 정확도를 감소시킨다. 또한, 공간 데이터 기반의 부하 제한 기법도 공간 데이터 스트림의 입력 빈도 변화와 공간 데이터의 밀집도를 반영하지 않아 질의 처리 정확도와 질의 처리 성능이 저하되는 문제가 존재한다. 이에 본 논문에서는 u-GIS 환경에서 부하 발생 빈도를 최소화하고 연속 질의 처리 성능과 정확도를 향상시키기 위해 공간 데이터의 밀집도와 공간데이터스트림의 입력 변화량을 이용하여 동적으로 부하를 제한하는 기법을 제안한다. 제안 기법에서는 부하제한 요구시 공간 이용도에 따라 질의에 참여할 확률이 낮은 데이터를 샘플링함으로써 연속 질의 처리 결과의 정확도와 질의 처리 속도를 향상시킬 수 있다.

**Abstract** In u-GIS environments, various load shedding techniques have been researched in order to balance loads caused by input spatial data streams. However, typical load shedding methods on aspatial data lack regard for characteristics of spatial data, also previous load shedding approaches on spatial, which still lack regard for spatial data density or dynamic input data stream, give rise to troubles on spatial query processing performance and accuracy. Therefore, dynamic load shedding scheme over spatial data stream is proposed through stored spatial data deviation and load ratio of input data stream in order to improve spatial continuous query accuracy and performance in u-GIS environment. In proposed scheme, input data which are a big probability related to spatial continuous query may be a strong chance to be dropped relatively.

**Key Words** : Data Stream, Load Shedding, Spatial Continuous Query, Smart Object

### 1. 서론

유비쿼터스 환경에서 위치 정보를 이용한 응용 서비스를 지원하기 위한 기반 기술로 u-GIS 플랫폼 기술이 대두되고 있다[1-2]. u-GIS 플랫폼 기술은 건물, 도로, 하천과 같은 2차원 또는 3차원상의 정적인 공간 정보와 유

비쿼터스 환경에서 시간에 따라 위치 정보가 포함된 동적인 GeoSensor 정보의 연계 처리가 요구된다. GeoSensor에서 생성되는 실시간 공간 정보를 처리하기 위해 데이터 스트림 처리 기술이 활용되고 있다[3-6]. 실시간으로 수집된 대용량의 데이터 스트림은 처리 과정에서 유한한 저장 공간을 초과할 수 있고 이로 인해 데이터가 손실되

이 논문은 2013년도 호서대학교 재원으로 학술연구비 지원을 받아 수행된 연구임(2013-0334)

\*Corresponding Author : Weonil Jeong(Hoseo Univ.)

Tel: +82-41-540-5984 email: wncung@hoseo.edu

Received October 29, 2014

Revised(1st December 12, 2014, 2nd January 13, 2015)

Accepted March 12, 2015

Published March 31, 2015

는 현상으로 인해 질의 처리 결과의 정확도가 저하될 수 있으므로, 이러한 문제를 해결하기 위한 부하제한 기법들이 연구되고 있다[7-13].

기존의 부하제한 기법은 임의의 부하제한 기법, 의미적 부하제한 기법, 그리고 샘플링 기법으로 분류된다[17]. 임의제한 기법은 데이터의 중요도를 고려하지 않기 때문에 빠른 수행이 가능하지만 이로인한 질의의 정확도가 저하될 수 있다. 의미적 부하제한 기법은 과부하가 발생하는 스트림 큐에서의 각 데이터에 대한 중요도를 고려한 우선 순위 기반의 부하제한 기법이다. 이에 반해 샘플링 기법은 통계적 연산을 기반으로 샘플링 비율을 산정하여 부하를 제한한다. 이러한 기존의 데이터 스트림에 대한 부하제한 연구[7-12]에서는 특정 공간 영역의 데이터를 대상으로 질의 처리가 요구되는 공간 데이터 스트림 처리에 있어 공간 데이터의 중요도를 반영할 수 없으므로 공간 연산에 대한 질의 정확도가 감소할 수 있다[13,18].

이에 공간 데이터에 대한 공간 질의의 공간적 이용도에 따라 질의에 이용될 확률이 낮은 데이터를 샘플링하는 차등적 샘플링을 통해 질의의 정확도를 향상시키기 위한 연구[18]가 수행되었으나, 이 연구에서는 부하제한을 실행함에 있어 입력되는 공간 데이터 스트림에 대한 유입률에 대한 변화량을 고려하지 않고 있다. 또한, 공간 데이터 스트림 큐에 불필요하게 가중되는 부하를 1차적으로 필터링을 수행한 후 과부하가 발생할 경우 공간 데이터의 중요도를 고려하여 추가적으로 부하를 제한하는 연구[13]에서는 스트림 큐에 발생 가능한 과부하를 예측할 수 없을 뿐 아니라 실시간으로 유입량이 변화하는 공간 데이터 스트림의 특성을 동적으로 반영하지 않는다.

따라서 본 논문에서는 u-GIS 환경에서 효율적인 부하 제한을 위해 질의 대상 영역을 공간 분할하여 구성한 그리드 해시 구조를 기반으로 공간 데이터스트림의 입력 빈도의 변화량과 공간 데이터의 분포 특성을 고려하여 동적으로 샘플링을 수행함으로써 공간적 특성을 고려한 부하제한 기법을 제안한다.

그리고 제안 부하제한 기법은 공간 분할 해시 구조를 적용함으로써 공간 연산의 처리 속도를 향상시킬 수 있고 특정 공간 영역에 대해 저장 데이터의 밀집도를 통해 산출된 공간 이용도와 실시간으로 입력되는 공간 데이터 스트림의 변화량을 이용해 동적인 부하제한을 통해 질의 정확도의 향상시킬 수 있다.

## 2. 관련연구

### 2.1 공간 데이터스트림 처리

GeoSensor에서 실시간으로 생성되는 대용량 공간 정보를 처리하기 위해서는 공간 슬라이딩 윈도우 집계 질의 등에 대한 공간 연산 처리를 포함한 공간 데이터 스트림 관리 기술이 요구된다[3-6]. 공간 데이터 스트림에 대한 윈도우 집계 질의는 공간 연속 질의에 집계 연산이 추가된 일반적인 슬라이딩 윈도우 집계 질의[3-5]의 술어에 공간 연산이 포함된 형태이다.

공간 데이터 스트림 관리 시스템은 지형이나 지물들에 대해 기 구축된 공간 데이터베이스와 GeoSensor에서 생성되는 위치 정보가 포함된 공간 데이터 스트림을 처리하여 유비쿼터스 환경의 다양한 응용들을 지원한다. 공간 데이터를 지원하기 위해 오라클이나 MySQL은 공간 데이터를 관계형 데이터베이스로 지원한다[14-15]. 공간 데이터에 대한 표준화 및 사양에 대한 정의는 OGC에서 제공하고 있다[16].

### 2.2 부하제한 기법

공간 데이터 스트림은 실시간으로 끊임없이 생성되어 공간 데이터 스트림 관리 시스템에 유입되는 특성을 가지므로 입력 데이터를 위한 할당 메모리가 초과될 수 있다. 이에 공간 데이터 스트림 관리 시스템에서는 저장된 스트림 데이터의 일부를 버리는 부하 제한 기법을 적용하고 있다. 이러한 부하제한은 입력 데이터를 제한함으로써 질의 처리의 정확도는 감소시킬 수 있으므로, 어떤 방법으로 부하를 제한하느냐에 따라 정확도에 많은 영향을 미친다. 기존에 연구된 부하 제한 기법은 임의의 부하제한 기법, 의미적 부하제한 기법, 샘플링 기법으로 분류한다[17].

대표적인 샘플링 기법인 Brian 기법[10]은 샘플링 비율을 산정하기 위해 입력 데이터의 속도나 튜플 처리 속도 등을 고려하여 임의의 샘플링을 수행하지만, 공간적인 특성을 고려하지 않고 있어 공간 데이터 스트림에 대한 질의 처리의 정확도나 성능이 감소할 수 있다.

UGLD 기법[18]은 공간 영역의 데이터에 대한 가중치 기반의 공간 이용도에 따른 차등적 샘플링을 수행함으로써 공간 영역의 특성을 반영하고 있으나 동적으로 변화되는 공간 데이터스트림에 대한 부하제한에는 한계가 있다.

PFPLS 기법[13]에서는 공간 데이터 스트림에 대한 부하 제한 기법으로 선-필터링을 통해 스트림 큐에 불필요하게 가중되는 부하를 1차적으로 제한하며, 과부하 발생 시 공간 질의 결과 정확도를 보장하기 위하여 공간 중요도와 데이터 중요도를 고려하여 후-부하제한을 수행한다. 그러나 이 기법에서는 실시간으로 입력되는 공간 데이터 스트림의 유입량에 대한 변화를 고려하지 않으며, 스트림 큐에 대한 부하 발생 예측을 하지 않고 있어 이중적인 필터링으로 인한 부하 제한의 효율이 저하될 수 있다.

### 3. 공간이용도에 따른 동적 부하제한

제안 기법은 데이터스트림을 공간영역으로 분할하여 공간 데이터의 분포와 비중을 산출하고, 각 영역별로 저장된 튜플수와 데이터 중요도를 관리하며, 해당 공간 영역의 중요도를 기준으로 동적 샘플링을 수행하여 부하를 제한한다.

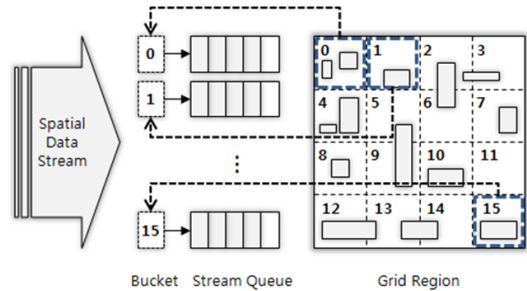
따라서 공간 접근이 빈번한 영역에 가중치를 두어 상대적으로 샘플링을 적게하여 질의 정확도를 향상시킬 수 있고, 데이터가 많이 입력된 영역에 샘플링을 증가하여 데이터 양에 따라 샘플링을 균등하게 하도록 한다. 또한 연산 공유시 공간 샘플링을 우선 적용하게 되어 특정 공간 위주로 샘플링을 하면 공간 연산을 사용하지 않는 질의는 편중된 데이터에 영향을 받아 질의 정확도가 감소될 수 있기 때문에 공간, 비공간 연산을 모두 고려하여 부하제한 연산자의 위치를 결정하여 정확도 감소를 방지할 수 있다.

#### 3.1 공간데이터스트림 관리구조

위치 정보를 포함하는 데이터 스트림을 공간 특성에 따라 영역별로 관리하기 위한 구조는 아래 [그림 1]과 같다.

Fig. 1에서 공간 데이터 스트림은 공간 정보를 갖는 객체와 조인 또는 공간 연산이 포함된 질의로 처리되는데, 이를 위해 질의 대상 영역을 그리드 셀로 분할한다. 분할 개수는 사용자가 임의로 설정할 수 있으며, 이 그리드 분할 수만큼 해당 영역의 데이터 스트림을 저장할 큐를 생성한다. 버킷(Bucket)은 분할된 그리드 영역에 대응되는 큐로써, 버킷 식별자(BID),  $i$ 번째 큐에 저장된 튜플 개수( $N_i$ )와 부하정도( $Li$ )로 구성된다.  $Li$ 는 단위 시간에 해당 큐에 입력되는 튜플의 변화량을 기반으로 해당 질의 공

간 영역에 대한 부하 정도를 산출한 값이며,  $Li$ 가 높아지면 버킷에 연계된 그리드 영역에 대한 공간 객체 참조 및 공간 연산 비율이 증가함을 의미한다.



[Fig. 1] Spatial Data Stream Structure

#### 3.2 질의 공간 부하율

질의 대상 공간 영역에 대한 부하 정도는 시간에 유동적인 데이터 스트림의 입력 빈도 변화율과 공간 연산 선택도의 산출을 통해 계산된다.

입력 데이터 스트림  $D_x$ 에 대해 데이터 스트림 생성 빈도의 변화율  $Freq_v(D_x)$ 는 아래와 같이 표현할 수 있다.

$$Freq_v(D_x) = Avg(Freq_v^{p's}(D_x)) + (|D_x^p - D_x^c|) \quad (1)$$

수식 1에서  $Avg(Freq_v^{p's}(D_x))$ 는 데이터 스트림의 변화 정도를 산출함에 있어 과거 입력된 데이터 스트림의 변화량을 평균값으로 계산한 예측값에 대한 보정값이며,  $(|D_x^p - D_x^c|)$ 는 데이터 스트림의 직전 입력 빈도와 현재 입력 빈도에 대한 절대 수치를 나타낸다. 이에 데이터 스트림의 입력 빈도  $Freq(D_x)$ 는 아래와 같이 유도된다.

$$Freq(D_x) = Freq_v(D_x) + D_x^c \quad (2)$$

수식 2에서 데이터 스트림  $D_x$ 에 대한 입력 빈도  $Freq(D_x)$ 는  $D_x$ 의 생성 빈도 변화율  $Freq_v(D_x)$ 와 현재 입력량  $D_x^c$ 의 합으로 유도된다.

공간 데이터 스트림은 연속 질의의 연산자에 의해 처리되며, 공간 선택도는 입력 데이터 스트림에 대해 연산자에 의해 처리된 중간 결과물의 비율을 나타낸다. 연산자에 입력되는 데이터 스트림도 시간에 따라 변화하며, 연산자  $Op_x$ 에 대한 선택도 변화율  $S_v(Op_x)$ 는 아래와 같다.

$$S_v(O_{p_x}) = |S^u(O_{p_x}) - S^c(O_{p_x})| + Avg(S^{p's}(O_{p_x})) \quad (3)$$

수식 3에서  $|S^u(O_{p_x}) - S^c(O_{p_x})|$ 는 연산자의 과거 선택도와 현재 선택도의 차이를 나타내며, 는 최근의 선택도 변화를 반영하는 변화율의 보정값을 의미한다.

$$S(O_{p_x}) = S_v(O_{p_x}) + S^c(O_{p_x}) \quad (4)$$

수식 4에서는 연산자  $O_{p_x}$ 에 대한 선택도  $S(O_{p_x})$ 는 해당 연산자에 대한 선택도 변화율과 현재 선택도를 합산하여 유도된다.

질의 공간 영역에 대한 부하 정도는 데이터 스트림 입력 빈도  $Freq(D_x)$ 와 연산자에 대한 선택도  $S(O_{p_x})$ 를 주기적으로 측정하여 산출한다.

$$L = \sum_{i=1}^n Freq(D_x) \cdot S(O_{p_x}) \quad (5)$$

수식 5에서 특정 공간 영역에 대한 부하율은 할당된 연산자의 수가 n개일 때, 각 연산자의 선택도와 데이터 스트림 입력 빈도에 대한 곱의 합으로 계산된다. 이는 부하율이 질의 공간 영역에 대한 공간 데이터 스트림의 입력 변화량 및 공간 연산의 곱에 비례한다.

이로부터 Fig. 1의 버킷 부하 정도(Li)를 산출한다. Li는 최대 k이하의 자연수이고 기본 증가 단위로 U를 사용하고, 이 값들은 임의로 결정할 수 있다. 또한 공간 객체가 각 그리드 셀에 포함되거나 접친 횟수를 GNi라고 하고 그 중 가장 큰 값을 M이라 한다. Li의 결정은 기본적으로 설정된 U를 사용하지만, M값에 따라 U의 값을 변경할 수 있다.

$$U = \left\lceil \frac{M}{k} \right\rceil, \text{ if } kU < M \quad (6)$$

$$L_i = \left\lceil \frac{GN_i}{U} \right\rceil, M = Max(GN_i) \quad (7)$$

수식 6은 부하율의 증가 단위로 U를 변경하는 수식이다. 위 식과 같이 M이 kU보다 크게 되면, 기본 U단위로 Li를 계산하고 최대 Li의 값인 k를 초과할 수 있으므로 U의 범위를 M/k로 변경 한다. 수식 7은 L을 구하는 수식으로, 각 그리드 셀의 GNi에 U로 나눈 값의 올림으로 Li을 구한다.

### 3.3 동적 공간 샘플링

제안 샘플링 기법은 각 저장된 튜플 수와 해당 큐와 연계된 공간 질의 영역에 대한 부하율을 이용하여 공간 샘플링을 수행한다. 부하율이 높다는 것은 해당 공간 영역에 존재하는 공간 데이터가 조인이나 연산에 참여할 확률이 높고, 입력 빈도의 증가는 소수의 튜플이 질의 결과에 미치는 영향은 상대적으로 감소함을 의미한다. 이에 해당 공간 영역에 대응하는 튜플 수인 공간 밀집도와 부하율에 비례하는 샘플링 비율을 동적으로 결정한다. 각 질의의 기본 샘플링 비율은 Brian[12]에 따른다.

튜플 수만으로 적절한 샘플링 비율을 분배하려면 공간 영역별로 P만큼 샘플링을 수행하면 영역 별로 동일한 비율로 분배된다. 여기에 공간 영역별 부하율을 적용한 샘플링 비율을  $P_k^t$ 라 할 때, 공간 영역별 적용 비율의 변경에도 일정한 샘플링을 위해서는 수식 8을 만족해야 한다.

$$\sum_{k=1}^N (1 - P_k^t) N_k^t = (1 - P_i) S \quad (8)$$

수식 8을 만족하면서 부하율이 높은 그리드 셀의 샘플링 비율을 낮추기 위해 다음과 같이 각 셀 별로 부하율이 적용된 튜플 수인 을 사용한다.

$$N_k^t = 0, \text{ if } L_k = 0 \\ N_k^t = N_k^t \times (1 - \alpha L_k), \text{ otherwise} \quad (9)$$

수식 9는 부하율이 적용된 튜플 수인  $N_k^t$ 을 결정하는 방법이다. 부하율이 0인 경우는 공간 연산에 이용되지 않으므로  $N_k^t$ 을 0으로 한다. 0이 아닌 경우에는  $(1 - \alpha L_k)$ 을 튜플 수에 곱하는데  $\alpha$ 는 부하율별로 샘플링 비율을 정하는 값으로  $\alpha$ 값이 높아질수록 부하율이 샘플링을 비율에 미치는 영향이 크게 된다.

각 공간 영역에 대한 부하율은 적용 튜플 수가 결정되면 아래와 같이 해당 샘플링 비율을 구할 수 있다.

$$\text{if } S = \sum_{k=1}^N N_k^t \text{ and } S_1 = \sum_{k=1}^N N_k^t, \\ P_k^t = 1 - \frac{N_k^t(1 - P_i)S}{S_L N_k^t} \quad (10)$$

## 4. 성능분석

### 4.1 평가환경

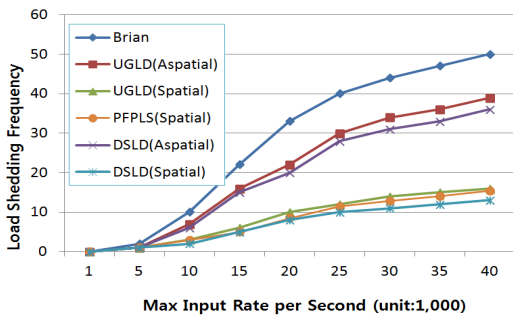
실험 시스템은 펜티엄 4 3.0 GHz, 4GB 메모리이며, 512MB 메모리를 시스템에서 할당하여 실험한다. 실험 데이터는 u-GIS 환경에서의 GeoSensor 데이터와 유사한 이동체 데이터를 생성하는 IBM City Simulation을 이용한다 [19]. 이 프로그램은 동시에 2백만 개의 이동객체를 생성할 수 있다. 공간 데이터는 TIGER/Line 2007 데이터를 상용 DBMS인 오라클 형식으로 저장하여 실험한다[20]. 두 종류의 데이터를 이용하여 다수의 공간 및 비공간 질의 처리를 통해 성능을 분석한다. 이동 객체를 저장하는 큐는 10MB의 크기로 초기에 할당하였다. 이동객체의 스키마는 <ID, Time, X, Y>이고 크기는 16바이트이다. 이동객체는 초당 지정된 최대 크기까지 임의의 수로 생성된다. 그리고 이동객체 데이터를 생성하는 데이터 스트림은 10개를 사용한다.

### 4.2 성능평가

평가 대상은 제안 기법(DSLD)과 Brian[12]과 UGLD[18], PFPLS[23]이고, u-GIS 환경에서의 사용자 질의에 따른 연산은 공간 연산과 비공간 연산으로 구분하여 실험한다.

#### 4.2.1 부하제한 발생 빈도

부하제한 발생 빈도 실험에서는 공간 질의와 비공간 질의를 조합하여 수행시키고 공간 데이터스트림의 입력수를 증가시키면서 부하제한 연산이 발생하는 빈도수를 측정한다.



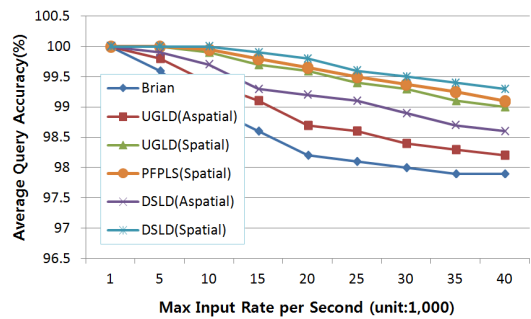
[Fig. 2] Load shedding frequency by increasing input data rate

Fig. 2에서 비공간 연산 공유 시 초당 입력이 10,000인 지점부터 Brian보다 DSLD(Aspatial)은 평균 26%,

UGLD(Aspatial)은 평균 33%의 부하제한이 적게 발생하고 있다. 그리고 공간 연산을 공유하는 경우에는 Brian 기법의 부하 발생이 다른 기법보다 상대적으로 더 크다는 것을 알 수 있으며, DSLD(Spatial)는 UGLD(Spatial)보다 평균 18%, PFPLS(Spatial)보다 11%의 부하제한이 적게 발생함을 확인할 수 있다.

#### 4.2.2 질의 정확도

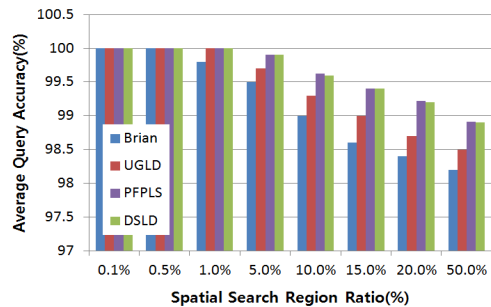
이 실험에서는 공간 데이터스트림의 입력 증가에 따른 질의 처리 결과의 정확도를 측정한다.



[Fig. 3] Average query Accuracy by increasing input data rate

Fig. 3은 비공간 질의에 비해 공간 질의가 많을수록 질의 처리 결과의 정확도가 높음을 알 수 있다. Brian에 비해 UGLD(Aspatial)은 0.4%, DSLD(Aspatial)은 0.8%의 높은 질의 정확도를 보여주고 있다. 또한 DSLD(Spatial)에서의 정확도가 UGLD(Spatial)에 비해 0.2%, PFPLS(Spatial)에 비해 0.12%가 높다. 이러한 결과는 부하제한 발생 빈도수에 증가하면 질의 처리 결과의 정확도가 저하됨을 의미한다.

다음 실험은 공간 질의 영역의 검색 범위 변화에 따른 질의의 정확도를 측정한다.



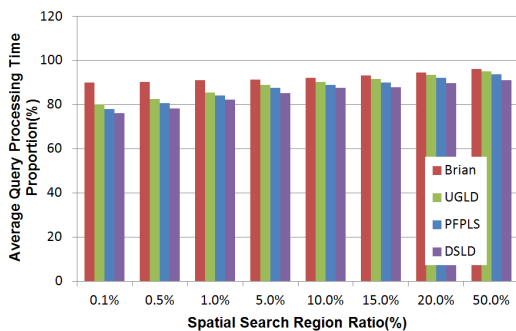
[Fig. 4] Average query accuracy by increasing spatial search region requested on spatial query

Fig. 4에서 실험 결과 전체적으로 공간 질의의 공간 검색 범위가 증가할수록 질의의 정확도는 떨어진다. 공간 검색 영역 비율 1%에서 Brian은 99.8%의 질의 정확도를 보이지만 다른 기법들은 100%를 유지하고 있는데, 이는 부하제한이 발생하지 않았기 때문으로 분석된다. 공간 검색 영역 1% 이후에서는 평균적인 질의 정확도가 Brian에 비해 DSLD가 0.59%가 높음을 알 수 있다. UGLD보다 DSLD가 높은 질의 정확도를 보이는 것은 공간 이용도에 따른 차등적 부하제한을 수행하는 UGLD보다 저장된 공간 데이터 및 공간 데이터스트림의 입력 빈도 변화량을 고려한 DSLD가 부하제한이 더 효과적이라는 것을 나타낸다. 그리고 제안 DSLD보다 PFPLS가 0.013% 차이로 나은 결과를 보이는 이유는 공간 데이터 스트림에 포함된 이동객체의 위치 변화량이 크지 않았기 때문으로 분석된다.

#### 4.2.3 질의 처리 성능

이번 실험에서는 비공간 연산이 공유될 때 공간 영역의 검색 범위 증가에 따른 질의 처리 성능을 비교한다. 부하제한이 이뤄지지 않을 경우 질의 처리 속도를 100으로 했을 때 비교 기법간의 질의 처리 속도를 비교한다.

Fig. 5에서는 비공간 연산에 연결된 공간 연산의 공간 검색 영역 비율이 증가할 때 Brian보다 UGLD는 4%, PFPLS는 5%, DSLD는 8%의 빠른 성능을 보이고 있다. 비공간 연산이 공유되면 연산 처리 시간이 증가할 수 있으나 공간 연산 처리에 질의 처리 시간이 이를 상쇄할 수 있으므로 검색 비율이 작은 경우에도 큰 영향은 없다. 그리고 부하제한 발생 빈도 실험의 결과로부터 UGLD와 PFPLS가 DSLD보다 더 많은 부하 제한이 발생하였으며 이에 따라 DSLD가 평균 처리 시간이 향상된 것으로 분석된다.



[Fig. 5] Average query processing time proportion by increasing spatial search region

## 5. 결론

본 논문에서는 u-GIS 환경에서 실시간으로 끊임없이 입력되는 공간 데이터 스트림에 대해 메모리 저장 공간의 한계를 해결하기 위한 효율적인 부하제한 기법을 제안하였다. 제안 기법은 공간 데이터 스트림의 입력 빈도 변화와 함께 저장 공간 데이터의 밀집도를 이용하여 부하 정도를 산출하고 샘플링을 수행하였다.

제안 부하제한 기법 평가는 비공간 데이터만을 고려하는 샘플링 기법인 Brian과 공간 데이터의 특성을 반영한 UGLD 및 PFPLS를 대상으로 수행하였다. 그 결과 제안 기법이 부하제한 발생 빈도, 공간 연산의 질의 처리 속도와 질의 결과의 향상된 정확도를 확인하였다. 향후 연구로는 부하분산의 효율성을 높이기 위해 다양한 센서에서 생성되는 데이터 패킷을 분석하여 중복 발생 데이터를 필터링하고 데이터의 유입시기를 설정하는 선처리에 대한 연구가 필요하다.

## References

- [1] C. Lee, K. An, M. Lee, J. Kim, "Trends of u-GIS Spatial Information Technology", ET Trends, Vol. 22, No. 3, pp.110-123, 2007.
- [2] K. An, J. Kim, "Design of Mobile u-GIS Information Processing System", Proc. of 30th KIPS Fall Conference, Vol. 15, No. 02, pp. 315-317, November, 2008.
- [3] Babcock, B., Babu, S., Datar, M., Motwani, R. Widom, J., "Models and Issues in Data Stream Systems", PODS, 2002.  
DOI: <http://dx.doi.org/10.1145/543613.543615>
- [4] Abadi, D. J., Carney, D., Centintemel, U., Cherniack, M., Convey, C., Lee, S., Stonebraker, M., Tatbul, N., Zdonik, S., "Aurora: A New Model and Architecture for Data Stream Management", VLDB Journal, 2003.  
DOI: <http://dx.doi.org/10.1007/s00778-003-0095-z>
- [5] Golab, L., Tamer Ozsu, M., "Issues in Data Stream Management", SIGMOD Record. ACM. Vol.32, No. 2, pp. 6-14, 2003.  
DOI: <http://dx.doi.org/10.1145/776985.776986>
- [6] H. Kang, C. Park, D. Hong, K. Han, "Development of a Spatial DSMS for Efficient Real-Time Processing of Spatial Sensor Data", Journal of KSIS, Vol. 9, No. 1, pp. 45-57, 2007.
- [7] Tatbul, N., Cetintemel, U., Zdonik, S., Cherniack, M., and

- Stonebraker, M., "Load shedding in a data stream manager", VLDB, Vol. 29, pp. 309-320, 2003.
- [8] Motwani, R., Widom, J., Arasu, A., "Query processing, approximation, and resource management in a data stream management system", CIDR, 2003.
- [9] Reiss, F. and Hellerstein, J., "Data Triage: An Adaptive Architecture for Load Shedding in TelegraphCQ", ICDE, 2005.  
DOI: <http://dx.doi.org/10.1109/ICDE.2005.44>
- [10] Babcock, B., Datar, M., and Motwani, R., "Load Shedding for Aggregation Queries over Data Streams", ICDE, 2004.  
DOI: <http://dx.doi.org/10.1109/ICDE.2004.1320010>
- [11] C. Basaran, K. Kang, Y. Zhou, M. Suzer, "Adaptive Load Shedding via Fuzzy Control in Data Stream Management Systems", SOCA, 5th IEEE International Conference, pp.1-8, Dec. 2012.  
DOI: <http://dx.doi.org/10.1109/SOCA.2012.6449438>
- [12] T. Pham, P. Chrysanthis, A. Labrinidis, "Self-managing load shedding for data stream management systems", ICDEW, IEEE 29th International Conference, pp70-76, Apr. 2013.  
DOI: <http://dx.doi.org/10.1109/ICDEW.2013.6547429>
- [13] H. Kim, S. Baek, D. Lee, G. Kim, H. Bae, "Pre-filtering based Post-Load Shedding Method for Improving Spatial Query Accuracy in GeoSensor Environment", Journal of KSIS, Vol. 12, No. 1, pp.18-27, 2010
- [14] "Oracle Spatial User's Guide and Reference 10g Release 1 (10.1)", Part No. B10826-01, www.oracle.com, 2003.
- [15] "MySQL 6.0 Reference Manual", www.mysql.com, 2008.
- [16] "OpenGIS Implementation Specification for Geographic information - Simple feature access - Part1:Common Architecture", www.opengeospatial.org, 2008.
- [17] Ho Kim, Sung-Ha Baek, Dong-Wook Lee, Gyoung-Bae Kim, Hae-Young Bae, "Load Shedding Applying Range Overlap Ratio of Spatial Query over Data Stream", ASGIS 2009, pp. 49-55, 2009.
- [18] S. Baek, D. Lee, G. Kim, W. Chung and H. Bae, "Load Shedding Method based on Grid Hash to Improve Accuracy of Spatial Sliding Window Aggregate Queries", Journal of KSIS, Vol. 11, No. 2, pp.89-98, 2009.
- [19] Kaufman, J., Myllymaki, J., and Jackson, J., "City Simulator", Alpha Works Emerging Technologies, Nov. 2001.
- [20] "Tiger/Line Shapefiles", census.gov/geo/www/tiger/tgrshp2007/tgrshp2007.html, 2007.

정 원 일(Weonil Jeong)

[정회원]



- 1998년 2월 : 인하대학교 전자계산 공학과(공학사)
- 2004년 8월 : 인하대학교 컴퓨터정보공학과(공학박사)
- 2004년 7월 ~ 2006년 7월 : 한국 전자통신연구원 선임연구원
- 2007년 3월 ~ 현재 : 호서대학교 정보보호학과 교수

<관심분야>

데이터스트림, 이동객체, 시스템보안