데이터 스트림에서 공간의 이용도를 이용한 차등적 부하제한 기법

김호*, 백성하*, 이동욱*, 신숭선*, 배해영*
*인하대학교 정보공학과
e-mail : {hokim, shbaek, dwlee, hermit}@dblab.inha.ac.kr, hybae@inha.ac.kr

Different Load Shedding using utilization of Spatial over Data Stream

Ho Kim*, Sung-Ha Baek*, Dong-Wook Lee*, Soong-Sun Shin*, Hae-Young Bae*
* Dept. of Computer Science and Information Engineering, Inha University

요 약

u-GIS 환경에서 GeoSensor 로부터 수집되는 시공간 데이터는 데이터 스트림의 특징을 포함한다. 데이터 스트림은 다양한 입력 속도로 끊임없이 입력되고, 데이터의 크기 또한 가변적이다. 이런 이유로 한정적인 메모리와 처리능력의 시스템은 과부하 현상이 발생한다. 이를 해결하기 위해 초과되는 데이터를 버려 메모리 초과를 방지하는 기법들이 연구되고 있다. 공간질의는 공간과 위치 값을 기반으로 이루어지는 연산으로 공간질의 정확도는 공간과 위치 정보를 통해 보장된다. 그러나 기존 기법인 랜덤부하제한 기법과 의미적부하제한 기법은 공간질의가 요구하는 공간과 위치 값에 대해 고려하지 않고 삭제하기 때문에 공간질의에 대한 정확도가 감소하는 문제를 갖는다. 본 논문에서는 공간의 이용도를 이용하여 차등적 비율을 적용한 부하제한 기법은 연구하였다. 이 기법은 등록된 공간질의의 영역 겹침 정도에 따라 중요 레벨을 증가시키고, 이를 토대로 시공간 데이터의 중요도를 파악하여 중요도마다 주어진 비율에 의하여 차등적으로 삭제한다. 결과적으로 기존 기법보다 다소 높은 Drop rate 를 통해 질의 처리 속도를 빠르게 회복시켰으며, 중요 데이터를 최대한 유지하여 Error rate 를 감소시켰다.

1. 서론

u-GIS 공간 정보 기술은 기존 GIS 인 건물, 도로, 하천, 지하시설물과 같은 2 차원 또는 3 차원 상의 정적인 지형 지물 정보와 유비쿼터스 환경을 기반으로 시간에 따라 공간적인 위치가 포함된 동적인 GeoSensor 정보의 융합 처리를 요구한다.[1]

GeoSensor 는 넓은 지역에 산발적으로 분포하며, 수집하는 데이터는 시간과 공간을 포함하는 것은 물론 지속적으로 발생하고, 가변적인 크기의 데이터 스트림의 성향을 갖는다.[2] 또한 수집되는 데이터의 양은 항상 유동적이며, 때에 따라 폭발적으로 발생하기도 한다. 이런 특성으로 시스템 상의 한정적인 메모리와 처리 능력에 의해 성능 저하 현상이 발생하고, 데이터가 손실이 되는 현상이 발생한다. DSMS 에 대한 대부분의 질의들은 질의 처리기에 의해 지속적으로 처리되어, 그 결과가 많은 어플리케이션으로 출력되길 기대한다.[3] 그러나 과부하로 인한 데이터 손실은 질의 결과의 신뢰도를 감소시키는 문제를 갖는다. 과부하 발생을 해결하기 위하여 랜덤, 의미적, 샘플링에 의한 부하제한기법 등 다양한 연구가 활발히 진행되고 있다.

랜덤부하제한 기법은 데이터 스트림 버퍼 내에 존재하는 데이터를 무작위로 선택하고 삭제하는 방법으로 빠른 처리 속도를 갖는다. 그러나 데이터의 중요도를 고려하지 않기 때문에 질의 결과의 신뢰도를 보장하지 못한다. 의미적부하제한 기법에 경우 데이터의 중요도를 반영하여 중요도가 낮은 데이터를 삭제하고, 중요도가 높은 데이터는 질의 처리될 수 있도록 한다. 따라서 질의 결과의 신뢰도 향상에는 많은 기여를 하였다. 그러나 공간질의에 경우 위치 값을 이용한 연산을 동반하는 만큼 공간의 특수성과 이용도를 고려해야 하지만 의미적부하제한 기법은 데이터의 중요도를 반영하기 위해 공간이나 위치 값을 대상으로 하지 않는다.

본 논문에서는 시공간 데이터 유입에 따른 과부하 발생을 해결하기 위해, 공간의 이용도에 따라 중요도를 부여하고 부여 받은 중요도에 따라 비율적으로 부하 제한하는 기법(Different Drop)을 제안한다. 이 기법은 기존에 연구한 공간질의의 영역 겹침을 이용한 우선순위 기반의 부하 분산 기법[2]을 확장 및 문제점을 보완하였다. 공간질의(연속질의 및 일회성질의)가 사용할 특정 공간에 대해 중요 레벨을 증가시킨다. 영역 겹침에 따라 증가되는 중요 레벨은 해당 공간의 중요 척도가 된다. 시공간 데이터의 위치 정보는 해

본 연구는 건설교통부 첨단도시기술개발사업 - 지능형국토정보기술혁신 사업과제의 연구비지원(07 국토정보 C05)에 의해 수행되었습니다.

당되는 영역의 중요 레벨에 의해 중요도가 판단되고, 데이터의 중요도에 따라 삭제 여부를 결정하게 된다. 데이터 삭제 시 여러 질의에 대한 결과를 만족시키기 위하여, 무조건적 삭제가 아닌 중요도 기준에 따른 차등적인 비율을 적용하여 부하제한이 이루어진다.

이와 같이 본 기법은 공간적인 특성을 이용한 중요도 관리 테이블과 중요 레벨에 따른 차등적 비율을 적용한 테이블을 이용한다. 기존 기법과 랜덤부하제한 기법에 비하여 높은 Drop rate 에도 불구하고, 여러 질의들의 정확도를 향상시켜 Error rate 를 감소시켰다.

2. 관련연구

2.1. 차등적 부하 분산 기법(DLSM)

중요 영역에 대한 고려 없이 삭제하는 랜덤・의미적부하제한 기법은 위험을 초래할 수 있다. 예를 들면, "인천광역시 용현동 화재의심지역(온도센서의 측정값이 50℃ 이상)의 위치를 출력하라."와 같은 공간 연속질의가 처리될 때, 화재의심지역이라는 영역에 대한 중요도가 고려되지 않고 삭제되면 즉각적인 초동 조치가 어렵다. 그래서 기존에 연구한 공간질의의 영역 겹침을 이용한 우선순위 기반의 부하 분산 기법은 u-GIS 에서 발생할 수 있는 문제점을 해결하는데 목적을 두었다.[2]

기존 논문은 공간의 우선순위를 맵 테이블로 유지하고, 맵 테이블을 토대로 유입된 데이터의 위치 정보를 비교하여 우선순위를 부여하였다. 부여된 우선순위는 LevelCounter 를 통해 우선순위의 값만큼 증가되고 해당 Level 값과 동등할 때 하나의 데이터를 삭제하는 방법을 사용했다. 그러나 이 기법은 차등적 비율 설정 알고리즘이 빈약하여 과부하 발생 해소가 어려웠다. 부하제한 처리 속도 역시 질의가 요구하는 처리 시간과 데이터 크기, 유입된 데이터 양의 따라 차이를 보였다. 이에 본 논문에서는 기존 논문의 문제점을 보완하고, 향후 연구 과제로 남겼던 부분에 대해 연구하였다.

2.3. Aurora 부하제한 기법에 대한 3 가지 중점

Aurora DSMS(현재 Borealis Project 에 포함)에서의 부하제한 기법은 차등적인 QoS 를 전재로 이루어지고 있다. 이 부하제한을 결정하기 위해 3 가지 중점을 고려하며, 그 항목은 When, Where, How much 이다.[5] 먼저 When 은 시스템으로 유입되는 데이터의 양을 주기적으로 체크하여, 과부하가 발견된 시점에 결정된다. Where 와 How much 는 수행 가능한 부하제한 계획을 유지하는 LSRM(Load Shedding RoadMap)에서 선택된 계획에 따라 수행되는 지점과 부하제한이 되는 데이터의 양이 다르다. 시스템의 유용도 손실률을 최소화하기 위하여 최적의 계획을 선택하기 때문이다. Aurora 와 Borealis 에서의 부하제한 기법은 3 가지 항목에서 When 과 How much 보다 Where 에 대해 더 중점을 두고 있다.

본 논문에서는 Where 에 대해 데이터 스트림 버퍼에서 질의 처리기로 유입되는 지점이 최적의 위치라 판단하여 적용하였고, When 과 How much 에 대한 결정은 과부하 발생 예측과 발생시점, 비율에 따라 삭제될 데이터 양을 판단하는 것을 목표로 하였다.

3. Different Drop 기법

본 논문은 시공간 데이터의 특성을 최대한 이용하여, 공간의 중요도를 이용한 차등적 비율 부하제한 기법인 Different Drop 기법을 제안한다. 본 기법은 등록된 공간질의가 이용할 특정 공간에 대해 중요 레벨을 관리하는 중요도 관리 테이블(PMT)과 데이터 스트림 버퍼로 유입되는 데이터의 양을 체크하는 부하제한 모니터(LSM), 중요 레벨과 유입된 총 데이터 수에 따라 비율적으로 데이터 삭제 수를 결정하는 차등적 비율 관리 테이블(DRT)로 구성된다.

3.1. 중요도 관리 테이블(Priorities Management Table)

본 절에서는 등록된 공간질의들의 영역 겹침에 의한 중요도를 관리하는 테이블에 대해서 설명한다.

실제맵의 공간 영역을 가상맵으로 구축하며, 공간질의가 포함한 영역의 겹침 정도에 따라 중요도를 반영한다. 중요도는 부하제한 수행 시 데이터의 삭제 여부를 판단하는 근거가 되는 역할을 담당한다.

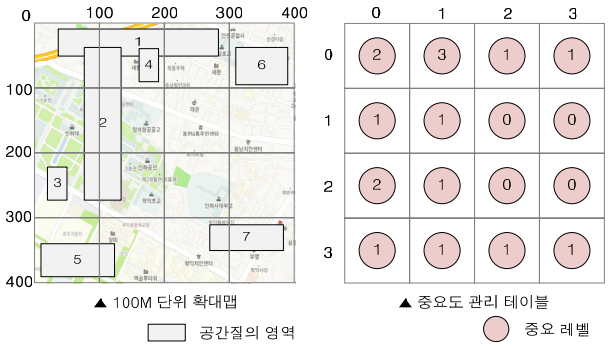
중요도 관리 테이블은 시스템 구동 시 초기화되며, 등록된 공간질의가 포함하는 영역에 대한 분석이 이루어진다. 이 때 (식 1)을 이용하여 질의가 포함하는 영역의 실제맵(Rx, Ry)을 가상맵(Vx, Vy)으로 맵핑하기 위한 비율을 산정한다.

$$ratioX = (MaxVx - MinVx) / (MaxRx - MinRx)$$
$$ratioY = (MaxVy - MinVy) / (MaxRy - MinRx) \quad (식\ 1)$$

산정된 비율 (ratioX, ratioY)를 토대로 (식 2)를 적용하여 질의가 포함하고 있는 영역 (Rx, Ry)에 대한 가상맵 (Vx, Vy)에 맵핑 작업을 진행한다.

$$Vx = Rx * ratioX, \quad Vy = Ry * ratioY \quad (식\ 2)$$

맵핑 작업은 Min(Vx,Vy)과 Max(Vx,Vy)에 의하여 이루어지며 [그림 1]에서 가상맵과 같은 공간질의 영역이 설정된다.



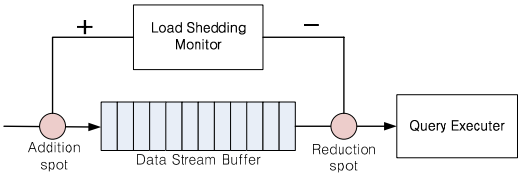
[그림 1] 중요도 관리 테이블(PMT)

설정된 영역은 PMT 에 중요 레벨을 증가시켜 해당 영역의 중요도를 반영한다. [그림 1]에서는 등록된 질의 q_1~q_7, 7 개의 질의를 분석하여 영역 설정이 되었으며, q_1, q_2, q_4 가 겹친 영역에 대해 중요 레벨 3 이 적

용되어 가장 높은 중요 영역인 것을 확인할 수 있다.

3.2. 부하제한 모니터(Load Shedding Monitor)

본 절에서는 데이터 스트림 버퍼로 유입되는 데이터의 수를 관리하는 부하제한 모니터를 설명한다. [그림 2]와 같이 LSM 은 데이터 스트림 버퍼마다 개별적인 모니터링을 실시한다. DRT 의 각 중요 레벨마다 적용된 비율에 맞게 보존해야 할 데이터의 수를 결정할 수 있도록 데이터 스트림 버퍼 내의 데이터 수를 전달하는 역할을 한다.



[그림 2] 부하제한 모니터(LSM)

LSM 은 데이터 스트림 버퍼 내에 유입되기 전 지점(Addition Spot)에서 데이터의 수를 증가시키고, 데이터 스트림 버퍼에서 질의 처리기에 입력되는 지점(Reduction Spot)에서 데이터 수를 감소시킨다. 또한 특정 GeoSensor 로부터 수집된 데이터의 형태가 일정하다는 특성을 고려해 데이터의 크기, 유입속도 등을 응용하여 과부하가 발생할 시점을 예측할 수 있다.

3.3. 차등적 비율 테이블(Different Ratio Table)

본 절에서는 질의 처리기에 입력될 데이터의 수를 결정하기 위한 차등적 비율 테이블을 설명한다. DRT 는 중요도가 낮은 데이터를 무조건 삭제하지 않고 그 일부 데이터들이 반영되어, 중요도가 높은 영역의 데이터만 편중 처리되는 것을 막기 위함이다.

중요 레벨은 PMT 로부터 최대값(p)을 부여 받아 중요 레벨이 가장 낮은 1(i)부터 차례대로 생성한다. 각 중요 레벨(PL_i)은 보존 데이터의 비율($Ratio_i$)을 산정하기 위하여 (식 3)을 이용한다. 보존 비율(PR)은 사용자에 의해 인자값으로 입력받으며, 기본값은 1 이다.

$$Ratio_i = PR - PL_i \Big/ \sum_{j=0}^{p} PL_j, \quad i \,\&\, j \le p \qquad \text{(식 3)}$$

이 때 PMT 에서 새로운 질의가 추가 등록되어 영역 겹침에 따른 중요 레벨의 최대값이 증가되었을 경우, DRT 역시 즉각적으로 갱신 작업이 이루어진다.

시스템 수행 중 과부하가 발생하면 LSM 으로부터 데이터 스트림 버퍼에 유입된 총 데이터 수(DataTotCnt)를 할당 받는다. DRT 는 총 데이터의 수를 (식 4)에 적용하여 비율에 비례하게 각 중요 레벨($Ratio_i$)마다 보존해야 할 데이터 수(PC_i)를 계산한다.

$$PC_i = Ratio_i * DataTotCnt \qquad \text{(식 4)}$$

[그림 3]은 유입된 데이터의 수가 250 개로 가정하고 과부하가 발생하였을 때, 중요 포인트가 최대 5 인 경우와 7 인 경우의 DRT 생성 모습이다.

▼ 최대 5개의 영역 겹침, 데이터 총 개수: 250

Priority Level	Ratio	Preservation #
1	0.06	15
2	0.13	32
3	0.20	50
4	0.26	65
5	0.33	83

▼ 최대 7개의 영역 겹침, 데이터 총 개수: 250

Priority Level	Ratio	Preservation #
1	0.03	7
2	0.07	17
3	0.10	25
4	0.14	35
5	0.17	42
6	0.21	52
7	0.25	62

[그림 3] 차등적 비율 테이블(DRT)

3.4. Different Drop 알고리즘

본 절에서는 제안한 PMT, LSM, DRT 를 이용하여 Different Drop 기법의 수행과정을 설명한다.

```
Algorithm DifferentDrop()
Input    없음
Output   없음
Begin
01:   nCnt := 0
02:   DataTotCnt := LSM()
03:   DRT.SetPresentationData(DataTotCnt)
04:   while nCnt < DataTotCnt
05:      CheckRec := RecvRecord(nCnt)
06:      X := CheckRec.X
07:      Y := CheckRec.Y
08:      PriorityLevel := PMT(X, Y)
09:      if DRT(PriorityLevel) then
10:         DeleteRecord(CheckRecord)
11:      end if
12:      nCnt := nCnt + 1
13:   end while
End
```

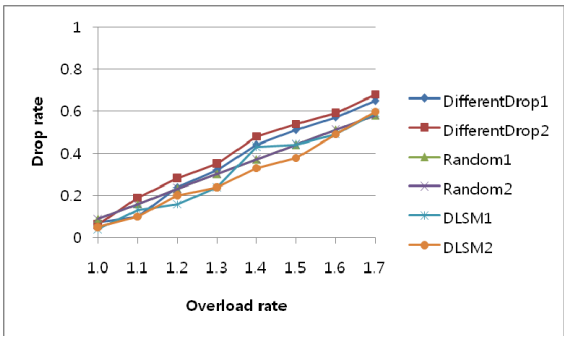
과부하가 발생할 시 DifferentDrop 을 실행하여 부하제한 수행한다. DifferentDrop 은 줄 2 에서 LSM 으로부터 데이터 스트림 버퍼 내에 존재하는 데이터의 총 개수를 할당 받는다. 할당 받은 DataTotCnt 토대로 줄 3 에서 DRT 의 각 중요 레벨의 보존할 데이터 수를 산정한다. 줄 5 에서 버퍼 내에 존재하는 데이터를 받고, 줄 6,7 에서는 중요도를 판별하기 위해 할당 받은 데이터의 위치 정보를 해당 변수에 저장한다. 줄 8 에서 데이터는 중요도를 부여 받기 위해 PMT 에서 위치 정보를 토대로 (식 2)를 이용하여 접근한다. 부여 받은 중요도는 줄 9 에서 DRT 로 보내지며, 테이블 내에서 해당되는 중요 레벨의 보존 데이터 수를 감소시킨다. 각각의 중요 레벨의 보존 데이터 수가 감소하여 0 이 되면, 이후에 진입된 데이터는 모두 부하제한 대상이 된다. 주어진 DataTotCnt 만큼 데이터의 삭제 여부 검토 후 DifferentDrop 은 정상 종료한다.

4. 성능평가

테스트에 사용된 시스템 환경은 Intel Pentium 4 CPU 3.00GHz, 4GB Memory 로 구성되어 있으며, 설치된 운영체제는 Fedora 9.0 이다.

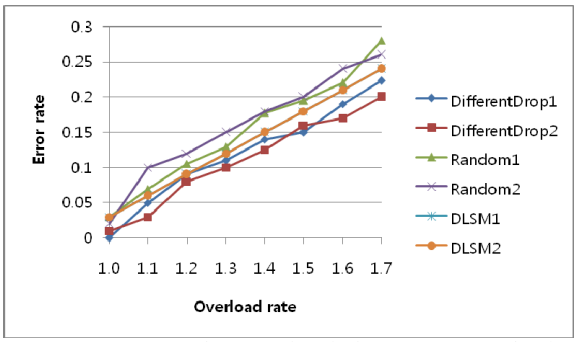
GeoSensor 에서 수집되는 데이터는 JAVA 를 이용하여 시뮬레이터를 구현하였으며, 데이터는 생성 시간과 위치 정보를 포함, 남은 필드의 값은 랜덤으로 생성하되 Max 사이즈는 28byte 로 제한하였다. GIS 데이터는 TIGER/Line 2007 를 Oracle 10g 에 구축하여 가상 맵 데이터를 확보하였다.

구축된 시뮬레이터와 가상맵을 기반으로, 질의 등록은 7 개로 설정하되, 영역 겹침에 의한 최대 중요 레벨은 '5'로 설정하였다. DRT 의 총 비율은 기본값인 1(100%)로 설정하였다. 실험 대상 기법은 본 기법인 Different Drop 기법과 기존에 연구한 DLSM, 랜덤부하제한 기법으로 선정하였으며, 좀 더 정확한 측정을 위하여 2 회씩 테스트하였다. 발생되는 데이터는 중요도와 관계 없이 임의적으로 발생하며, Drop rate 와 Error rate 를 측정하였다.



[그림 4] Overload Rate 따른 Drop Rate 측정

[그림 4]는 특정 Overload rate 에서 데이터를 삭제되는 비율을 측정한 결과이다. 랜덤부하제한 기법에 경우 데이터 유입량에 따라 부하제한 되는 비율이 일정하게 증가하고, DLSM 기법은 랜덤부하제한 기법보다 다소 낮지만 역시 일정한 비율로 데이터를 삭제한다. 그러나 Different Drop 기법은 다른 기법보다 다소 높은 Drop rate 로 측정되었다. 이는 질의처리 속도를 빠르게 회복시키는 현상을 보였다. 또한 버퍼 내에 존재하는 데이터들의 중요도에 따라 과부하 발생 시점에서 차등적 비율을 적용하여 삭제하기 때문에 같은 Overload rate 에서도 다른 Drop rate 를 보인다.



[그림 5] 중요 레벨 5 에 대한 Error Rate 측정

[그림 5]는 시뮬레이터에서 발생한 데이터의 수와 질의 처리기에 진입한 데이터의 수에 차이 정도를 측정한 것으로 중요 레벨이 '5'인 데이터들을 대상으로 한 결과이다. 랜덤부하제한 기법은 측정 대상인 중요 레벨 '5'인 데이터를 고려하지 않기 때문에, 일정하지 않은 Error rate 를 보인다. DLSM 에 경우는 5 개의 데이터를 질의 처리기에 삽입하고 하나의 데이터를 삭제하기 때문에 일정한 Error rate 를 갖는다. 반면 Different Drop 은 다른 두 기법보다 높은 Drop rate 에도 불구하고 낮은 Error rate 를 보인다. 중요도마다 보존해야 할 비율을 달리하여 중요도가 높은 데이터에 대해 보다 많이 질의 처리될 수 있도록 했기 때문이다. 또한 다른 기법과 달리 Error rate 가 일정하지 않은 이유는, 매 테스트마다 중요 레벨이 '5'인 데이터가 데이터 스트림 버퍼에 적재되는 수가 다르고, 크기 역시 유동적인 이유로 버퍼 내 총 데이터의 수가 달라졌기 때문이다.

5. 결론

u-GIS 환경에서 발생되는 시공간 데이터는 공간이라는 특수성이 내포되어 중요도가 낮은 데이터일지라도 무조건 삭제는 위험 처사이다. 따라서 공간질의의 영역 겹침을 통하여 영역의 중요도를 결정하고 그에 따라 차등적 비율을 적용하여 부하제한 하는 Different Drop 기법을 수행하였다. 실험을 통해 중요도가 낮은 데이터일지라도 질의 처리에 반영되었고, 중요도가 높은 데이터들이 보다 많이 질의 처리될 수 있도록 하였다. 그 결과 랜덤부하제한 기법과 DLSM 기법에 비하여 다소 높은 Drop rate 를 보였지만, 두 기법보다 낮은 Error rate 를 보이면서 질의처리 속도를 빠르게 회복시켰다.

향후 연구로는 대표적 상황을 설정하여 부하제한 계획을 수립하고, 부하제한 모니터를 통해 데이터의 유입량, 유입속도, 크기를 고려하여 과부하 발생시점을 예측한다. 그에 따라 최적의 부하제한 계획을 선택하여 시스템 상의 안정화와 성능 향상에 대한 연구를 진행할 것이다.

참고문헌

[1] 백성하, 이동욱, 김경배, 정원일, 배해영, "공간 슬라이딩 윈도우 집계질의의 정확도 향상을 위한 그리드 해쉬 기반의 부하제한 기법," 한국공간정보시스템학회 논문지 제 11 권 1 호, 2009.

[2] 김호, 백성하, 이연, 이동욱, 정원일, 배해영, "데이터 스트림에서 공간질의의 영역 겹침을 이용한 우선순위 기반의 부하 분산 기법", 제 30 회 한국정보처리학회 추계학술대회 제 15 권 2 호, 2008.

[3] Y. Tu, S. Liu, S. Prabhakar, and B. Yao, "Load Shedding in Stream Databases: A Contorl-Based Approach," In VLDB Conference, Seoul, Korea, 2006.

[4] N. Tatbul, S. Zdonik, "Window-Aware Load Shedding for Aggregation Queries over Data Streams," In VLDB Conference, Seoul, Korea, 2006.

[5] N. Tatbul, U. Cetintemel, and S. Zdonik, "Load Shedding in a Data Stream Manager," In VLDB Conference, Berlin, Germany, 2003.

[6] N. Tatbul, Y. Ahmad, U. Cetintemel, Jeong-Hyon Hwang, Y. Xing, and Zdonik, "Load Management and High Availability in the Borealis Distributed Stream Processing Engine," Springer-Verlag Berlin Heidelberg, 2008.

[7] B. Babcock, S. Babu, M. Datar, R. Motwani, and J.Widow, "Model and Issues in Data Stream System," Proc. of ACM PODS, 2002.

[8] B. Babcock, M. Datar, and R. Motwani, "Load Shedding for Aggregation Queries over Data Stream," Proc. of the 20th ICDE, 2004.