DOI: 10.3745/KIPSTD.2009.16-D.4.507

짧은 구간을 갖는 범위 질의의 효율적인 질의 색인 기법

김 재 인[†] · 송 명 진[†] · 한 대 영[†] · 김 대 인[††] · 황 부 현[†††]

요 약

데이터 스트림 환경에서는 지속적으로 입력되는 데이터에 대한 실시간 처리를 수행하기 위하여 범위를 갖는 다수의 질의를 시스템에 미리 등록한다. 등록된 질의를 입력 스트림에 따라 빠르게 검색하기 위해 질의 색인 기법을 사용하는데, 질의 색인은 메인 메모리 기반에서 동작하기 위해 색인 정보의 저장 비용이 낮아야 하고 빠른 질의 탐색을 실시해야 한다. 본 논문에서는 다수의 범위 질의에 대하여 색인 정보의 저장 비용이 낮고 빠른 질의 탐색을 실시하는 질의 색인 기법으로 LVC-based(Limited Virtual Construct-based) 기법을 제안한다. 해시기반으로 동작하는 LVC-based 색인 기법은 입력 스트림의 범위를 가상의 분할 구조로 나눈 LVC를 이용한다. 각 LVC는 식별자가 할당되고 각 LVC에 구간에 해당하는 범위 질의를 저장하며 색인을 실시한다. LVC-based 기법은 입력 스트림의 범위가 길고 범위가 짧은 다수의 질의를 색인할 때 저장비용과 탐색 비용에서 좋은 효율을 보이며 이는 기 제안된 CEI-based 색인 기법과의 비교를 통하여 입증하였다.

키워드 : 데이터 스트림, 범위 질의, 질의 색인, 연속 질의

Efficient Query Indexing for Short Interval Query

Jaeln Kim[†] · MyungJin Song[†] · DaeYoung Han[†] · Daeln Kim[††] · BuHyun Hwang[†††]

ABSTRACT

In stream data processing system, generally the interval queries are in advance registered in the system. When a data is input to the system continuously, for realtime processing, a query indexing method is used to quickly search queries. Thus, a main memory-based query index with a small storage cost and a fast search time is needed for searching queries. In this paper, we propose a LVC-based(Limited Virtual Construct-based) query index method using a hashing to meet the both needs. In LVC-based query index, we divide the range of a stream into limited virtual construct, or LVC. We map each interval query to its corresponding LVC and the query ID is stored on each LVC. We have compared with the CEI-based query indexing method through the simulation experiment. When the range of values of input stream is broad and there are many short interval queries, the LVC-based indexing method have shown the performance enhancement for the storage cost and search time.

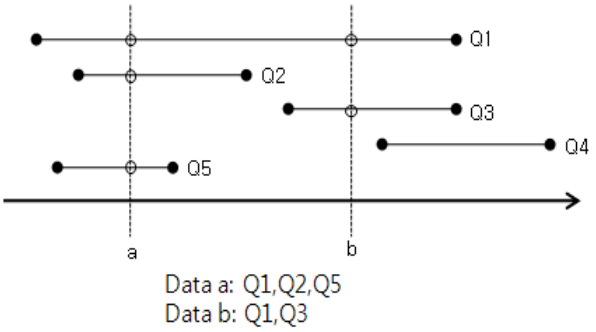
Keywords : Data Stream, Interval Query, Query Indexing, Continuous Query

1. 서 론

최근 금융 분야의 다양한 응용, 네트워크 통계, 통신 데이터 관리, 다양한 센서를 포함하는 센서 네트워크와 같은 스트림 데이터에 대한 다양한 응용들이 연구 되고 있다[1][2][8]. 스트림 데이터 환경에서, 시스템은 수집된 데이터를 실시간으로 처리하여 그 결과를 사용자에게 제공해야 한다. 이러한 스트림 데이터의 실시간 처리를 위하여 질의를 미리 시스템에 등록하고 입력 스트림에 해당하는 질의를 탐색하여 연속적으로 질의(continuous query)를 수행한다[12][13]. 이와 같이 연속적으로 수행되는 다수의 질의를 효율적으로 저장하고 입력 스트림에 대한 질의를 빠르게 탐색하기 위하여 질의 색인(query indexing) 기법을 사용한다[3][12][13].

스트림 데이터 환경에서의 질의는 특정 구간을 조건으로 갖는 특징을 가진다. 예를 들어 온도 정보를 수집하는 센서 네트워크에서 온도가 30℃~35℃ 사이인 경우 알람을 울리도록 정의할 수 있는데 이와 같은 특정 조건에 해당되는 범위를 포함하는 질의를 범위 질의(interval query)라고 한다[3][9]. 질의의 범위는 질의의 술어(predicate) 부분에 기술되며 수집되는 스트림 데이터가 등록된 질의의 범위에 해당하는 경우에만 해당 질의를 수행하도록 해야 한다. 범위 질의의 탐색은 (그림 1)과 같이 등록된 다수의 범위 질의에 대하여 입력 데이터의 값에 따라 질의의 조건을 만족하는 해

※ 본 연구는 본 논문은 2009년도 학술진흥재단 기본연구지원사업 (2009-0076136)에 의하여 연구 되었음.
† 준 회 원 : 전남대학교 전자컴퓨터공학부 석사과정
†† 정 회 원 : 전남대학교 전자컴퓨터공학부 시간강사
††† 종신회원 : 전남대학교 전자컴퓨터공학부 교수(교신저자)
논문접수 : 2009년 4월 8일
수 정 일 : 1차 2009년 5월 25일, 2차 2009년 6월 16일
심사완료 : 2009년 6월 16일



(그림 1) stabbing query

당 질의를 모두 찾는 stabbing query[9] 처리와 유사하다. (그림 1)은 5개의 범위 질의가 시스템에 미리 정의되어 있으며 데이터 *a*와 *b*가 수집된 예를 보여 준다. 데이터 *a*는 질의 *Q1*, *Q2*, *Q5*의 범위 조건에 해당하며 데이터 *b*는 질의 *Q1*, *Q3*의 범위 조건에 해당되며 해당 질의는 각각의 데이터를 사용하여 수행되어 실시간으로 데이터를 처리한다.

다수의 범위 질의가 등록된 시스템에서 연속적으로 수집되는 스트림 데이터를 처리하기 위해 해당 질의를 검색하기 위한 질의 색인 기법은 해당 질의를 빠르게 검색하여야 하며 메인 메모리상에서 빠르게 동작하기 위하여 낮은 저장 비용을 갖도록 해야 한다. 이러한 질의 색인 기법의 예로 트리 방법 기반의 IBS-trees[6], IS-lists[4][7], interval-trees[3], R-trees[5] 방법이, 그리고, 해시 방법 기반의 VC-based (Virtual Constructs-based) 기법과 CEI-based(Construct Encoded Interval-based)[10] 기법이 제안되었다.

트리 기반 색인 기법은 이진 탐색을 통하여 빠른 질의 탐색이 가능하지만 범위 질의 처리에는 부적합하며 질의 등록 및 삭제 시 트리 재구축 비용이 발생하는 문제가 있다[7][10]. 해시 기반 색인 기법은 가상 분할 영역 VC(Virtual Construct)를 사용함으로써 탐색 및 질의 등록/삭제 등에서 좋은 성능을 보이지만 가상 분할 영역의 크기나 개수에 따라 저장되는 색인 정보가 증가하는 단점을 보인다. 특히 시스템에 정의된 질의의 범위가 매우 큰 경우와 입력 스트림의 범위가 큰 경우 저장 공간의 효율이 떨어지는 문제가 있다[10].

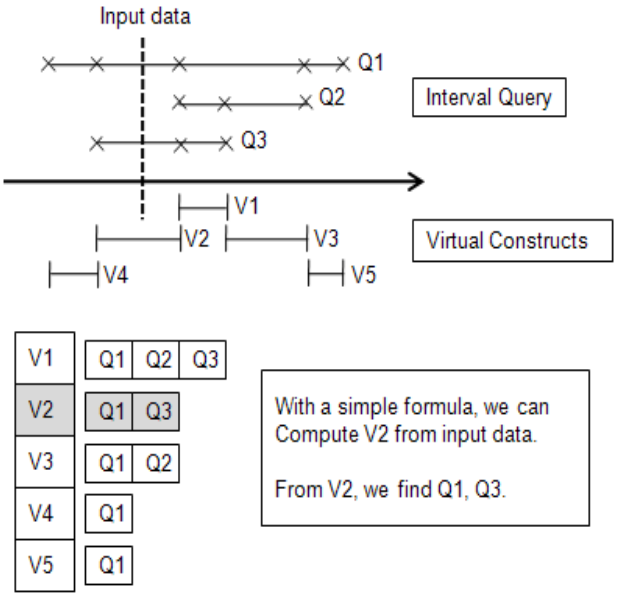
본 논문에서는 효율적인 스트림 데이터 처리를 위한 질의 색인 기법으로 해시 기반의 LVC-Based 색인 기법을 제안한다. 제안하는 기법은 기존의 CEI-based 기법에 비하여 수집되는 스트림 데이터의 범위가 광범위 하더라도 우수한 검색 성능 및 저장 비용의 효율성을 보인다. 본 논문의 구성은 다음과 같다. 2장에서는 본 논문과 관련된 가상 분할 구조에 대한 VC-based, CEI-based 기법에 대하여 알아보고, 3장에서는 제안하는 LVC-based 기법을 소개한다. 4장에서는 성능 비교를 실시하고, 끝으로 5장에서는 결론 및 향후 연구를 기술한다.

2. 관련 연구

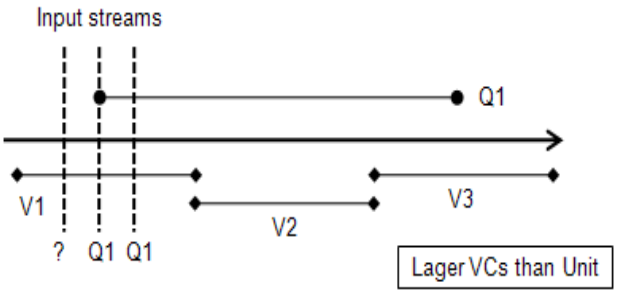
빠른 질의 탐색을 위한 색인 기법으로 제안된 방법들은 크게 두 가지로 분류 된다. 첫 번째는 이진 탐색을 이용한 트리 기반 색인 기법이고 두 번째는 해시 기반 가상 분할 구조 색인 기법이다. 트리 기반 색인 방법은 이진트리 탐색을 통하여 비교적 빠른 검색 성능을 보이지만 범위 질의 표현이 어렵고 질의 추가 및 삭제 시에 트리를 재배치해야 하는 문제가 있다[7][10]. 그러므로 일반적으로 스트림 데이터에 대한 범위 질의를 포함한 연속 질의 탐색을 위한 색인 기법은 해시 기반 방법이 사용되고 있다[10][11].

[10]에서는 해시 기반 질의 색인 기법으로 VC-based 기법과 CEI-based 기법을 제안하였다. VC-based 기법의 기본 아이디어는 센서를 통하여 수집되는 스트림 데이터의 범위를 가상 분할 영역인 VC로 분할하고 각각의 VC에 식별자를 부여하여 등록된 질의의 범위와 일치하는 VC의 식별자와 질의 식별자를 저장하여 (그림 2)와 같은 질의 색인 정보를 구성하는 것이다.

(그림 2)의 질의 색인에서는 3개의 질의와 5개의 VC가 정의 되어 있으며, 각 VC의 범위와 일치하는 질의에 대하여 각 VC에 질의 식별자를 저장한다. 그리고 입력되는 스트림 데이터에 대하여 해시 함수를 적용하여 해당되는 VC 식별자를 구하고 해당 식별자에 저장된 질의를 탐색한다. 하지만 (그림 2)와 같은 구조의 VC 정의는 등록되는 질의에 의하여 이루어지며 질의의 등록이나 삭제에 따라 VC의 영역을 다시 정의해야 하기 때문에 VC의 재정의 비용이 발생하므로 실시간 시스템에서는 부적합 하다. 따라서 VC의 크기를 미리 정의 한 후에 질의 색인을 실시하도록 한다. 정확한 질의 색인을 하기 위해서는 등록된 질의의 자료형의 기본 단위로 색인을 실시하는데 등록된 질의가 정수형 일 때, VC의 크기는 1이 된다. VC의 크기가 1인 경우 입력 스트림에 대하여 해시 함수를 적용하여 정확한 VC를 찾게 되고 해당 VC에 등록된 질의를 탐색한다. 하지만 VC의 크기가 1인 경우 긴 범위의 질의를 색인 하는 경우 여러 VC에 같은



(그림 2) VC-based 기법의 예

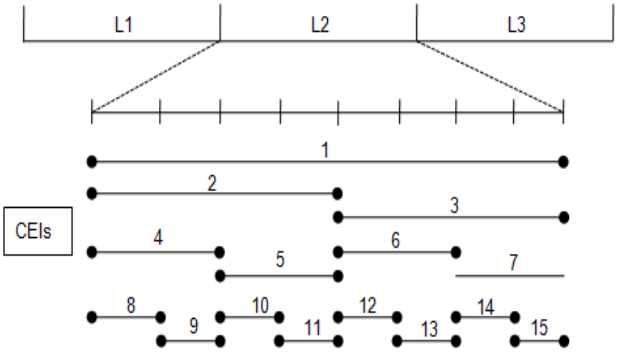


(그림 3) VC-based 기법의 문제점

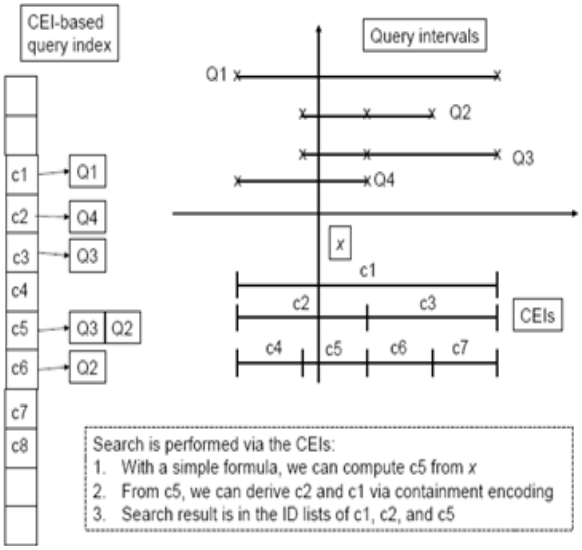
질의가 반복적으로 저장되어 저장 공간의 낭비를 초래 한다. 또한 1의 단위로 VC를 정의 하여 색인 하는 경우 입력 스트림의 범위가 넓고 구간이 짧은 범위 질의가 등록되는 경우 질의 정보를 저장하지 않는 다수의 VC가 저장되어야 함으로 저장 공간의 낭비를 발생시킨다. 이러한 단점을 보안하기 위해 VC의 크기를 크게 하여 색인 하는 경우 (그림 3)과 같이 질의를 색인 하는데 있어서 부정확한 색인 정보가 저장되어 정확한 질의 탐색을 실시하지 못하는 문제가 발생한다. 위와 같은 VC-based 기법의 중복질의 저장 문제, 부정확한 질의 탐색 등의 단점을 해결하고자 CEI-based 기법이 제안되었다[10].

CEI-based 기법은 VC-based 기법과 유사하게 (그림 4)와 같은 가상의 분할 영역인 CEI를 구성하여 범위 질의에 대한 색인 정보를 구축한다. 이 때 입력되는 스트림 데이터의 범위에 대한 가상 영역의 크기가 L인 경우 그 영역은 다시 $2L-1$개의 CEI로 분할된다. (그림 4)의 예에서 스트림 데이터의 범위에 대한 하나의 가상 영역 크기는 L_x=8로 정의되었고 하나의 가상 영역은 다시 15(2*8-1)개의 CEI로 분할되며 각각의 분할 영역에는 아이디가 할당된다.

CEI-based 기법에서 입력되는 스트림 데이터의 타입이 정수일 때 CEI의 크기는 정확한 질의 색인을 위하여 최하위 레벨의 CEI((그림 4)의 CEI_8~CEI_{15})의 경우 최소 정수 단위인 1의 크기로 정의되며 그 상위 레벨은 하위 레벨의 2배 크기로 계층적으로 정의된다. 이러한 구조를 갖는 이유는 입력된 스트림 데이터에 해당되는 CEI 영역 최하위 레벨의 CEI_x를 검색한 후 해당 CEI_x를 포함하는 상위 레벨의 CEI 영역을 이진 탐색으로 해당 데이터에 대한 범위 질의를 빠



(그림 4) 분할된 CEI의 예



(그림 5) CEI-based 기법의 예

르게 탐색하기 위함이다. 또한 계층적 CEI의 구조는 VC-based 기법에서 발생한 긴 범위 질의의 중복 저장을 최소화 한다. 왜냐하면 긴 범위의 질의는 최상위 레벨에 정의된 긴 구간의 CEI로 저장이 되고 짧은 범위의 질의는 하위 레벨에 정의된 짧은 구간의 CEI로 저장이 되기 때문이다.

(그림 5)는 이러한 CEI-based 기법을 이용한 질의 색인과 질의 탐색을 나타내고 있다. CEI-Based 기법에서는 먼저 입력 스트림의 범위에 따라 가상 분할 영역인 CEI를 정의한다. (그림 5)에서는 4개의 질의에 대하여 8개의 CEI가 정의 되어 있으며 각 CEI의 구간에 해당하는 질의를 저장함으로써 질의를 색인한다. 질의의 탐색 과정은 다음과 같다. (그림 5)에서는 입력 값 x에 대한 질의를 검색하기 위하여 먼저 입력 값 x에 해시 함수를 적용하여 해시 함수의 결과 값에 해당하는 최하위 레벨의 CEI인 $c5$를 찾는다. 검색된 $c5$로부터 상위 레벨의 CEI로의 검색을 실시한다. $c5$로부터 $c2$와 $c1$을 검색하게 되고 $c5$, $c2$, $c1$에 저장된 질의를 찾음으로써 질의 탐색을 마치게 된다.

긴 범위 질의의 중복 문제를 해결하고 빠른 탐색 시간을 제공하는 CEI-based 기법은 등록된 질의 수에 상관없이 전체 입력 스트림의 범위에 대하여 모든 CEI를 저장하고 있어야 하므로 스트림 데이터의 범위가 광범위하고 상대적으로 정의된 범위 질의가 적은 경우 색인 정보를 저장하는 비용의 낭비가 발생한다. 왜냐하면 CEI-based 기법에서는 (그림 5)의 $c4$처럼 질의 정보를 가지지 않는 CEI도 모두 저장하기 때문이다. 또한 범위 질의 탐색은 최하위 레벨의 CEI에서 상위 레벨의 CEI로 검색해야 하기 때문에 범위 질의의 구간이 충분히 짧은 경우 최하위 레벨의 CEI에 질의가 저장된다면 상위 레벨의 CEI로의 불필요한 탐색 비용이 발생하는 문제가 있다. 따라서 입력 스트림의 범위가 넓고 짧은 구간을 갖는 다수의 질의를 색인 하는데 있어서 색인 정보의 저장 공간을 최소화 하면서 빠른 탐색 시간을 유지하는 방법에 관한 연구가 필요하다.

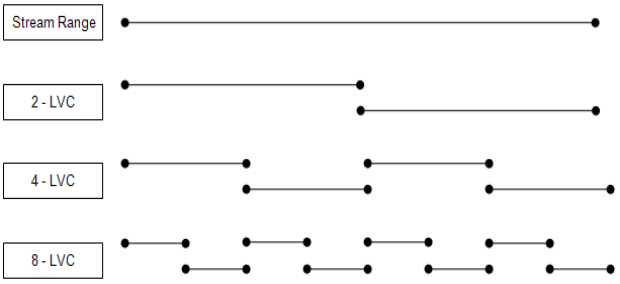
3. LVC-based 기법

3.1 LVC-based 기법

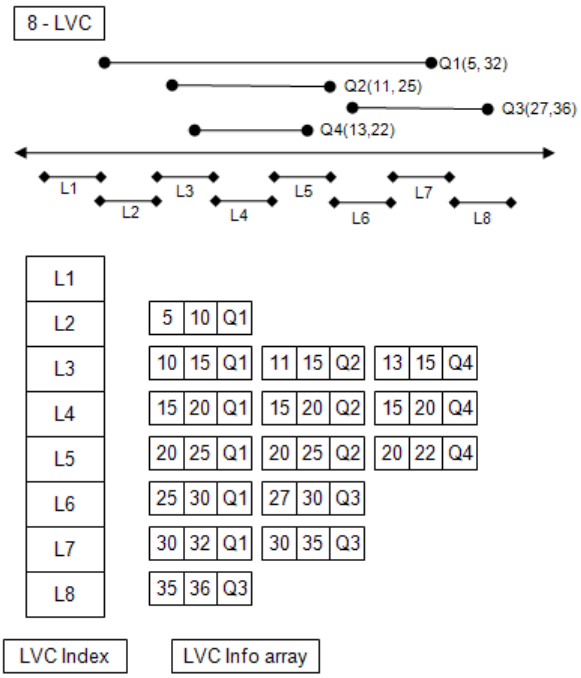
LVC-based 기법은 해시 기반 범위 질의 색인 기법으로 먼저 입력 가능한 스트림의 범위를 (그림 6)에서처럼 LVC라는 가상 분할 영역을 정의한다. CEI-based 기법에서 제안하는 가상 분할 영역은 등록된 다수의 범위 질의의 성격과 무관하게 일괄적으로 가상 분할 영역을 정의 한다. 이러한 가상 분할 영역에서의 색인은 등록된 다수의 범위 질의의 성격에 따라 탐색 시간 비용과 색인 정보 저장 공간 비용 면에서 비효율성을 제공한다. 예를 들어 등록된 다수의 범위 질의가 짧은 범위를 가지는 경우, 입력 스트림의 범위를 입력 스트림의 자료형의 최소단위로 영역을 분할하는 CEI-based 기법에서는 하위 레벨에서 상위 레벨로의 불필요한 탐색 비용이 발생하게 된다. 또한 등록된 범위 질의가 입력 스트림 공간에서 특정 부분에 집중되어 있는 경우 질의 정보를 포함하지 않는 다수의 CEI 정보를 유지해야 함으로 저장 공간 낭비의 문제점이 발생하게 된다. 따라서 제안하는 LVC-based 기법은 일괄적으로 가상 분할 영역을 정의 하면서 발생하는 문제점을 해결하기 위하여 응용 시스템의 목적에 맞는 범위 질의의 분포, 구간의 크기 등을 고려할 수 있도록 가상 분할 영역을 정의하도록 한다.

(그림 6)은 입력 스트림의 범위에 따라 다양하게 정의 될 수 있는 가상 분할 영역인 LVC를 표현한 것이다. 각 LVC는 시스템에 정의된 LVC의 전체 개수에 따라 일정 크기로 입력 스트림의 범위를 분할된다. 만약 등록되는 다수의 범위 질의가 큰 구간을 갖는 경우 각 LVC의 크기를 크게 정하고, 반대로 작은 구간을 갖는 경우 LVC의 크기를 작게 정의 한다. 이러한 정의는 응용 목적에 맞게 다양하게 정의될 수 있으며 등록될 범위 질의의 특성을 고려하여 효율적인 질의 색인을 가능하게 한다.

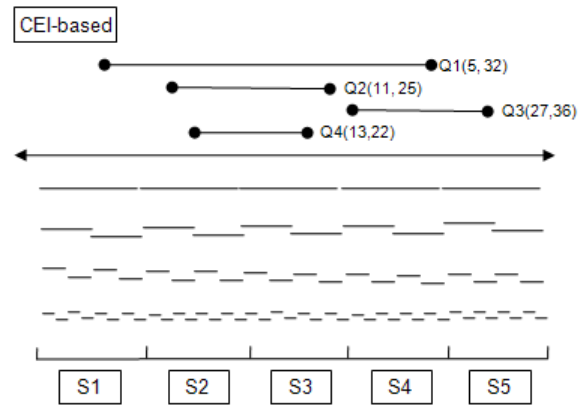
LVC-based 기법에서는 시스템에 정의된 LVC의 크기에 따라 CEI-based 기법에서 사용하는 분할 영역보다 보다 적은 수의 분할 영역을 사용하여 색인 정보를 저장한다. (그림 7)의 (a)는 4개의 질의에 대하여 LVC의 크기를 5로 하였을 때의 분할 영역과 분할 영역에 저장된 질의 정보를 나타내고 (b)는 같은 질의에 대하여 CEI-based 기법에서의 분할 영역을 나타낸다. 입력 스트림의 범위는 정수 0부터 40이기 때문에 LVC-based 기법에서는 8개의 분할 영역이 정의되



(그림 6) LVC 분할 예



(a) LVC-based 분할 영역 및 질의 색인



(b) CEI-based 분할 영역

(그림 7) 두 기법의 분할 영역

고 CEI-based 기법에서는 총 75개의 분할 영역이 정의 된다. 각 분할 영역에 질의 정보를 저장하더라도 LVC-based 기법에서의 질의 정보 저장 공간이 더 적게 소요됨을 알 수 있다.

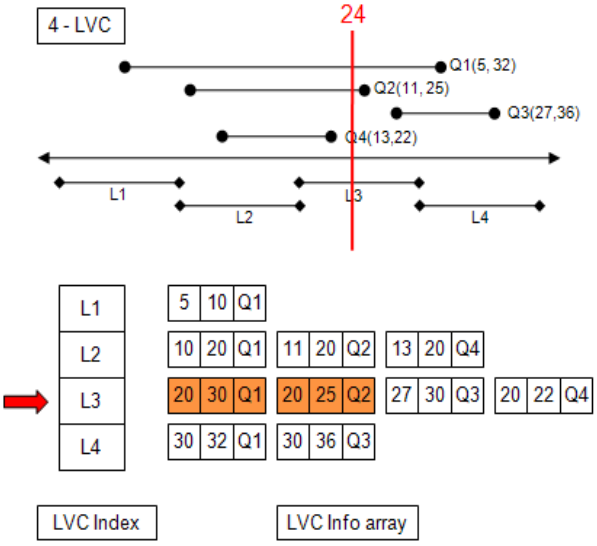
하지만 가상 분할 영역의 크기가 커짐으로써 (그림 3)에서처럼 분할 영역에 해당하는 범위 질의를 정확하게 저장할 수 없는 문제점이 발생 하게 된다. 이러한 문제점을 해결하기 위하여 LVC-based 기법은 각각의 LVC에 범위 질의 정보를 저장할 때 범위 질의의 구간 정보를 포함하는 *LVC_Info_array*라는 자료 구조를 정의하여 사용한다. *LVC_Info_array*는 범위 질의의 시작과 끝 정보와 해당 질의 식별자의 쌍으로 구성된다. 예를 들어 범위 질의 식별자가 3이고 범위 질의 구간이 $2 \leq x < 4$ 이면 식별자 3에 해당되는 범

위 질의 Q_3은 해당 LVC의 *LVC_Info_array*에 Q_3의 질의 범위에 대한 시작과 끝, 그리고 질의 식별자로 구성되는 *<2,4,Q3>* 정보를 저장한다. (그림 7)의 (a)는 4개의 범위 질의가 정의된 시스템에서 스트림 데이터의 전체 범위를 8개의 LVC로 분할하고 각 LVC의 *LVC_Info_array*에 해당 질의의 범위 정보를 저장한 예이다.

LVC-based 기법의 질의 정보 저장을 위한 자료 구조는 범위 질의 구간의 자료 형에 상관없이 저장 비용을 그대로 유지하고 구간이 긴 범위 질의의 중복 저장을 줄이는 장점을 갖는다. 예를 들어 입력되는 스트림 데이터의 자료 형이 실수 이고 등록된 범위 질의의 구간이 실수일 때, 기존의 VC-based 기법에서는 정확한 범위 질의 색인을 위하여 가상 분할 영역인 VC의 크기를 0.1로 정의해야 한다. 마찬가지로 CEI-based 기법에서도 가상 분할 영역의 최하위 레벨의 CEI의 크기를 0.1로 정의해야 한다. 이는 질의 색인 정보 저장 비용을 증가 시키며, 특히 구간이 긴 범위 질의에 대하여 많은 수의 VC나 CEI로의 중복 저장을 발생시킨다. 이와 같은 문제는 앞서 제시한 LVC의 질의 저장 자료구조인 *LVC_Info_array*를 통해 해결 할 수 있다.

3.2 질의 탐색 알고리즘

본 절에서는 LVC-based 기법에서의 질의 탐색 알고리즘을 기술한다. LVC-based 기법은 (그림 7)의 (a)와 (그림 8)과 같이 LVC의 크기에 따라 LVC의 개수와 *LVC_Info_array*에 저장되는 질의 수가 달라진다. 그러므로 LVC-based 기법은 등록된 질의의 특성을 분석하고 선처리하여 적절한 크기의 LVC를 정의함으로써 질의 탐색 성능을 향상 시킬 수 있다. 예를 들어 시스템에 등록된 범위 질의의 대다수가 짧은 범위를 갖는 경우 LVC 크기를 작게 하여 LVC의 개수를 증가시킴으로써 하나의 LVC에 대한 *LVC_Info_array*에 저장되는 질의 수를 줄일 수 있다. 또한 등록된 질의의 대다수가 긴 범위를 갖는 경우 LVC 크기를 증가시켜 LVC 개수를 줄



(그림 8) LVC-based에서 질의 탐색 예

임으로써 LVC에 저장되는 질의 범위 정보 크기를 줄일 수 있다.

LVC-based 기법에서 질의 탐색 과정은 먼저 입력되는 스트림 데이터에 대하여 해시 함수를 적용하여 입력된 스트림 데이터가 해당하는 LVC를 탐색한다. 해시 함수는 정의된 LVC의 범위에 따라 적절하게 구현되며 해시 함수를 적용한 후에 LVC의 식별자를 찾아 해당 LVC에 저장된 *LVC_Info_array*에서 질의를 탐색한다. 각 LVC의 *LVC_Info_array*에 저장된 범위 질의 구간을 입력된 스트림 데이터와 비교하여 입력 데이터에 해당되는 해당 범위 질의를 탐색한다. (그림 8)은 입력되는 스트림 데이터의 범위가 0부터 40 사이이며 4개의 질의가 정의된 스트림 데이터 시스템에서 LVC를 크기 10으로 구성한 경우 입력 데이터 24에 해당되는 범위 질의를 탐색하는 과정의 예이다.

(그림 8)에서 스트림 데이터 24가 입력되면 입력 데이터를 해시 함수에 적용하여 해당 LVC를 탐색한다. 예를 들어 (그림 8)에서 적용된 해시 함수는, LVC 크기가 10이고 입력 가능한 스트림의 범위는 0부터 40이므로 해시 함수 H(x) = ⌈x/10⌉가 되며 입력 데이터 24는(H(⌈24⌉)=3) LVC 식별자로 L3을 추출한다. 그리고 L3의 *LVC_Info_array*에 저장된 4개의 질의인 Q1, Q2, Q3, 그리고 Q4의 구간과 입력 데이터 24를 비교하여 입력 스트림 24에 해당되는 범위를 갖는 질의 Q1과 Q2를 탐색한다. 이러한 과정으로 수행되는 LVC-based 기법에서의 질의 탐색 방법은 알고리즘 1과 같다.

Input data : stream data
output data : Query ID

```
LI = LVC Index;
LIa = LVC_Info_array;
x = input data;
Search(x) {
  LI = H(x) // H(x) = ⌈x/ℓ⌉, ℓ=range of LVC
  for ( I=0; i<LIa[LI].size(); I++)
   {
    if ( LIa[LI][0] ≦ x < LIa[LI][1] )
        Result_Query += LIa[LI][2];
   }
  return Result_Query;
}
```

알고리즘 1. 질의 탐색 알고리즘

3.3 질의 등록 알고리즘

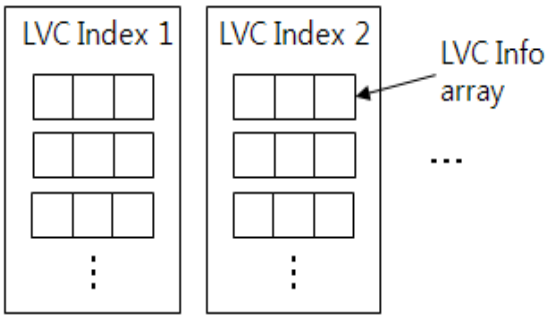
LVC-based 기법에서 질의 등록 과정은 LVC 크기의 정의로부터 시작된다. 즉, LVC-based 기법은 센서를 통하여

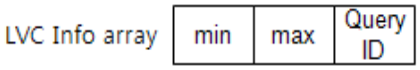
입력될 수 있는 스트림 데이터의 최대 최소 구간을 고려하여 정의된 LVC 크기에 따라 LVC 색인 정보를 저장하기 위한 기억 공간을 확보한다. 그리고 생성된 LVC는 질의 범위 정보를 저장하기 위한 *LVC_Info_array* 기억 공간을 (그림 9)와 같이 확보한다.

(그림 9)와 같이 LVC-based 기법에서 적용하는 LVC는 <*min*, *max*, *Query_ID*>와 같은 3차원 배열 형태로 구성된 *LVC_Info_array* 공간을 포함하며 이러한 공간은 새로운 질의를 추가할 수 있도록 동적으로 생성된다. 그리고 *LVC_Info_array*의 *min*에는 질의의 최소 범위값을, *max*에는 질의의 최대 범위값을, 그리고 *Query_ID*에는 질의 식별자를 저장한다. 질의를 시스템에 등록하는 과정은 정의된 LVC 범위에 따라 다음과 같은 세 가지 경우가 발생하는데 (그림 10)은 이러한 세 가지 경우에 대하여 등록되는 범위 질의와 LVC 간의 포함 관계를 보여준다.

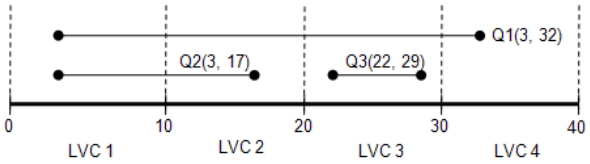
- Case 1 : 등록되는 범위 질의가 하나의 LVC에 포함되는 경우(*Q3*)
- Case 2 : 등록되는 범위 질의가 두 개의 LVC에 포함되지만 질의의 부분 범위와 정확하게 일치하는 LVC가 존재하지 않는 경우(*Q2*)
- Case 3 : 등록되는 범위 질의가 두 개 이상의 LVC에 포함되면서 질의의 부분 범위와 정확하게 일치하는 LVC가 존재하는 경우(*Q1*)

(그림 10)에서 범위 질의 *Q3*은 *LVC3*에 포함되므로 *LVC_Info_array*에는 <*22,39,Q3*>가 저장된다. 그리고 범위 질의 *Q2*는 2개의 LVC(*LVC1,LVC2*)에 포함되지만 Q2의 부분 범위 *R1*(3~10)과 *R2*(10~17)에 해당되는 LVC(*LVC1 (1~10)*과 *LVC2(10~20)*)는 존재하지 않으므로 *LVC1*과 *LVC2*의 *LVC_Info_array*에 각각 <*3,10,Q3*>과 <*10,17,Q3*>가 저장된다. 마지막으로 범위 질의 *Q1*과 같이 2개 이상의 LVC(*LVC1, LVC2, LVC3, LVC4*))에 걸쳐 질의가 포함되는 경우에는 질의의 부분 범위와 정확하게 일치하는 LVC가 존재하지 않는 영역인 *LVC1*와 *LVC4*에는 *Q2*와 같은 방법으로, 그리고 부분 범위와 정확하게 일치하는 LVC 영역인 *LVC2*와 *LVC3*에는 LVC 영역 값을 각 LVC의 *LVC_Info_array*에 저장한다. 이러한 과정으로 수행되는 LVC-based 기법에서의 질의 등록 방법은 알고리즘 2와 같다.





(그림 9) LVC 자료 구조



(그림 10) LVC와 질의 포함 관계

Input data : Interval Query
output data : LVC-based Index

```
Insert(min,max,query_id){
  i = ⌊min/ℓ⌋ = LVC Index // ℓ=range of LVC
  LR = ⌈min/ℓ⌉; LL = ⌊max/ℓ⌋;
  if (LL < LR)  // case 1
  { q_insert(min,max,query_id,i); }
  else {
    if (LR < LL) // case 3
    { q_insert(min,LR*ℓ,query_id,i);
      for (k=LR; k<LL; k++)
      { q_insert(k*ℓ,(k*ℓ)+ℓ,query_id,k); }
      q_insert(LL*ℓ,max,query_id,LL);
    }
    else // case 2
    { q_insert(min,LR*ℓ,query_id,i);
      q_insert(LR*ℓ,max,query_id,i+1); }
  }
}
q_insert(min,max,query_id,i){
    LVC Info array[0] = min;
    LVC Info array[1]= max;
    LVC Info array[2]= query_id;
    LVC Index[i]+= LVC Info array; }
```

알고리즘 2. 질의 삽입 알고리즘

LVC-based 기법에서 질의 삭제 또한 질의 등록과 유사하게 수행할 수 있다. 범위 a와 b를 갖는 질의 삭제는 질의 최소 범위값 a에 대하여 해시 함수 $H(a)$를 수행하여 해당 LVC를 계산한다. 그리고 질의의 최대 범위값 b에 대하여

해시 함수 $H(b)$를 수행하여 해당 LVC를 계산한다. 그리고 계산된 각각의 LVC에 대한 *LVC_Info_array*에 등록된 질의를 탐색하여 삭제한다. 이 때 질의 등록 과정의 Case 3의 경우에는 질의 삭제시에도 질의 최소 범위값 a에 대한 $H(a)$와 최대 범위값 b에 대한 $H(b)$부터 얻어지는 LVC 사이에 있는 모든 LVC를 포함하여 각각의 *LVC_Info_array*에 등록된 질의를 탐색하여 삭제한다.

4. 성능 평가

본 절에서는 제안한 LVC-based 기법과 [10]에서 제안한 CEI-based 기법과의 성능 평가를 수행한다. CEI-based 기법은 스트림 데이터 처리를 위한 해시 기반 질의 색인 기법으로 제안한 LVC-based 기법의 비교 대상으로 적합하다. 두 방법의 성능을 평가하기 위하여 입력되는 스트림 데이터의 범위 및 이에 기반한 범위 질의를 랜덤하게 생성함으로써 다양한 범위를 갖는 질의를 생성하여 수행한다. 성능 비교를 위하여 CPU 3.0GHz Dual, 2.0GB RAM 시스템을, 운영체제로 Windows XP SP3를 사용하였으며, 프로그램은 C++STL을 이용하여 작성하였다.

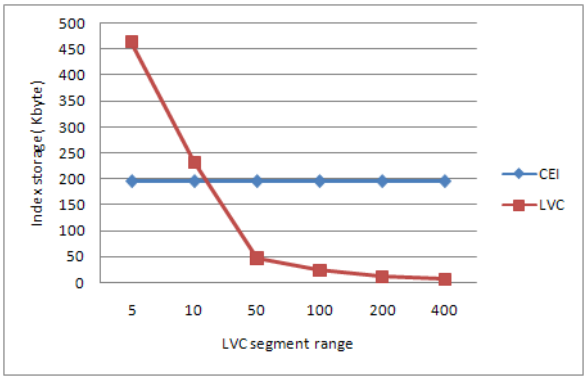
성능 평가에서는 시스템에 정의된 LVC의 크기에 따른 질의 색인 시뮬레이션을 통하여 탐색 성능 및 저장 비용에 대하여 CEI-based 기법의 성능과의 비교 분석을 실시한다. 먼저 질의 색인 정보의 저장 비용은 표 1과 같은 경우에 대하여 평가한다. 구간이 긴 범위 질의, 구간이 짧은 범위 질의, 특정 영역에 집중된 질의에 대하여 질의 색인을 실시하고 질의의 성격에 따른 저장 비용을 CEI-based 기법과의 비교를 통해 제안한 LVC-based 기법의 저장 공간 비용의 우수성을 입증한다. 시뮬레이션에서 색인할 범위 질의의 자료형은 정수형이고 입력 가능한 스트림 데이터의 범위는 0부터 10000으로 설정하였다. 그리고 <표 1>의 (3)의 경우인 특정 영역에 집중된 질의의 시뮬레이션에서는 질의의 구간을 4000~7000 사이로 정의하였다.

구간이 긴 범위 질의들을 색인 하는 경우 CEI-based 기법에서는 각 CEI에 질의가 중복되어 저장되므로 범위가 짧은 질의를 색인 하는 경우인 <표 1>의 (2)의 경우와 (3)의 경우 보다 더 많은 저장 공간이 필요하다. LVC-based 기법에서도 구간이 긴 범위 질의를 색인하는 경우, 각 LVC의 크기가 작을 때는 CEI-based 기법 보다 더 많은 저장 공간을 필요로 하지만 범위가 긴 질의의 성격에 맞게 각 LVC의 크기가 충분히 커지게 되면 저장 공간이 훨씬 더 적게 필요로 하는 것을 (그림 11)의 (a)를 통하여 알 수 있다. 구간이 짧은 범위 질의들을 색인 하는 경우에는 CEI-based 기법에서는 중복 저장된 질의의 수가 줄어들기 때문에 (1) 경우보다 더 작은 저장 공간을 필요로 하지만 LVC-based 기법의 각 LVC에 저장되는 중복된 질의의 수도 줄어들기 때문에 CEI-based 기법 보다 훨씬 더 작은 저장 공간을 필요로 한다. 또한 각 LVC의 크기가 충분히 큰 경우에는 중복 저장되는 질의가 발생 하지 않기 때문에 저장 비용은 더욱 낮아지게 된다. 특정 영역에 집중된 질의의 경우도 마찬가지로 범위 질의의 구간이 500 이기 때문에 (2)의 경우 보다 저장 공간을 조금 더 많이 차지 하지만 각 LVC의 크기가 커짐에 따라 작은 저장 공간을 필요로 하는 것을 알 수 있다. 이와 같이 LVC-based 기법은 범위 질의의 구간의 특성을 고려하여 각 LVC의 크기를 정의함으로써 질의 색인 정보의 저장 비용을 낮출 수 있음을 확인 하였다.

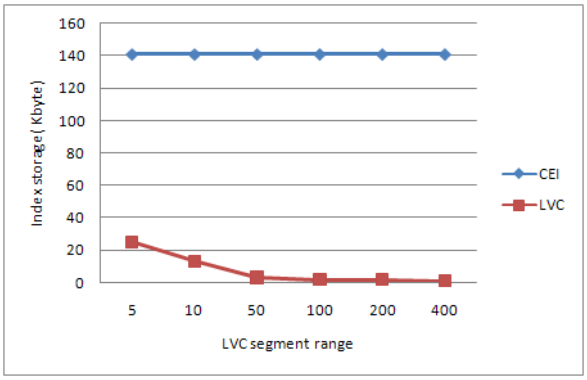
다음으로 질의 색인의 성능을 확인하는 가장 중요한 척도인 질의 탐색 시간에 대하여 성능 평가를 실시한다. LVC-based 기법에서의 각 LVC 크기에 따라 질의를 저장하기 위해 요구되는 저장 공간이 달라짐을 확인 하였다. 질의 탐색 시간 또한 각 LVC의 크기에 따라 그 결과가 달라진다. 그 이유는 LVC의 크기에 따라 하나의 LVC에 저장되는 질의의 수가 달라지기 때문이다. 이는 등록된 질의의 성격과 LVC의 크기에 따라 결정되며 등록된 질의의 특성에 맞게 LVC를 정의함으로써 좋은 탐색 성능을 보일 수 있다. CEI-based 기법에서의 질의 탐색은 최하위 레벨의 CEI로부터 최상위 CEI로의 질의 탐색을 실시함으로 등록된 다수의 질의가 짧은 구간을 갖는 경우 질의 정보가 저장되지 않은 CEI로의 불필요한 탐색을 실시하게 되어 탐색 비용이 증가하게 된다. LVC-based 기법과 CEI-based 기법의 탐색 성능 분석을 위하여 <표 1>의 (2) 경우와 같이 짧은 구간을 가지는 질의에 대하여 질의를 색인하고 저장된 색인 정보에 대하여 입력 스트림이 해당하는 질의를 탐색하는 시간을 시뮬레이션 한다. 시뮬레이션에서 범위 질의 구간의 자료형과 입력 스트림의 자료형은 정수형 이며 입력 스트림은 0과

〈표 1〉 질의 색인 정보 저장 비용 평가 환경

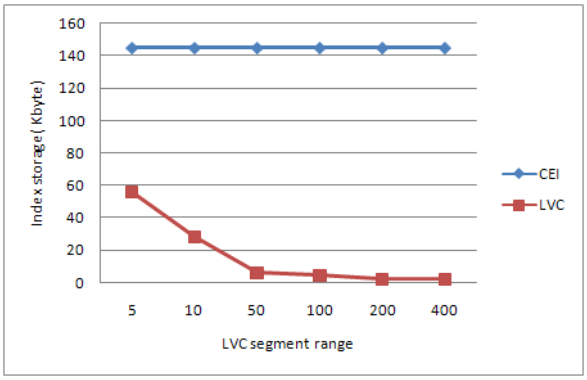
범위 질의	구간이 긴 범위 질의(1)	구간이 짧은 범위 질의(2)	특정 영역에 집중된 질의(3)
범위 질의 구간	0~3000	0~100	0~500
질의 개수	100		
스트림 범위	0 ~ 10000		



(a) 긴 구간 범위 질의 색인 저장 비용



(b) 짧은 구간 범위 질의 색인 저장 비용

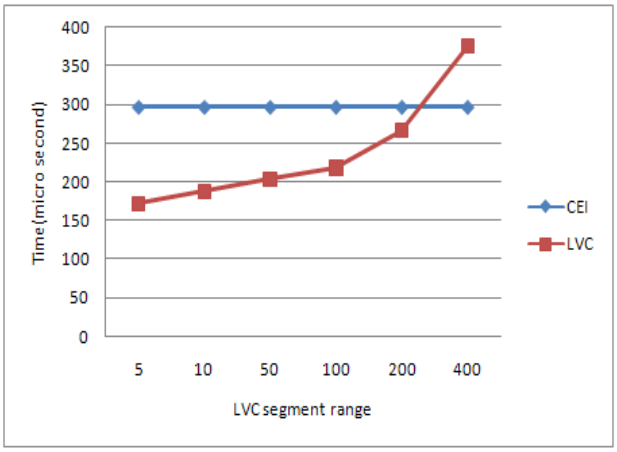


(c) 특정 영역의 범위 질의 색인 저장 비용

(그림 11) 질의 색인 저장 비용

10000 사이의 정수로 스트림 10^5개에 대한 질의 탐색 시간을 확인 한다.

(그림 12)는 시뮬레이션 결과를 나타낸 그래프로 LVC 크기에 따라 탐색시간이 달라지는 것을 확인 할 수 있다. LVC의 크기가 작으면 질의 색인을 위해 많은 저장 공간을 필요로 하지만 각 LVC에 저장되는 질의 정보가 적기 때문에 질의를 탐색 하는 시간이 줄어들게 된다. 하지만 저장 비용을 줄이기 위해 LVC의 크기를 키우게 되면 결국 각 LVC에 저장되는 질의가 많아지고 질의를 탐색하는 비용이 커지게 된다. 따라서 등록된 질의의 성격에 맞는 LVC 크기



(그림 12) 질의 탐색 시간

의 정의를 통해 질의 색인의 정보의 저장 비용과 질의 탐색 시간 비용을 최소화 할 수 있다.

질의 색인의 저장 비용과 입력 스트림에 대한 질의 탐색 시간 비용을 시뮬레이션을 통해 확인 해 보았다. 짧은 구간을 가지는 범위 질의에 대한 색인을 실시하는 경우, CEI-based 기법 보다 LVC-based 기법이 저장 공간과 질의 탐색 시간의 비용에서 정의된 LVC의 적절한 크기에 따라 더 나은 성능을 보이는 것을 확인 하였다.

5. 결론 및 향후 연구

본 논문에서는 스트림 데이터 처리를 위한 범위 질의 색인 기법으로 LVC-based 기법을 제안하였다. LVC-based 기법은 해시 기반 질의 색인 기법으로 가상 분할 공간인 LVC를 사용하여 기존 방법들에 비하여 저장되는 색인 정보 크기를 최소화하고 입력되는 스트림 데이터의 범위가 광범위한 경우, 보다 빠른 질의 검색이 가능하다. 또한 LVC-based 기법은 스트림 데이터의 범위나 시스템에 등록된 질의 수의 변화에도 유연하게 적용될 수 있다. 향후 연구로 스트림 데이터 처리의 효율을 보다 극대화하기 위하여 LVC의 크기를 능동적으로 결정할 수 있는 알고리즘에 대하여 연구하고자 한다. 또한 LVC에 포함된 질의의 범위 정보를 보다 효율적으로 저장함으로써 보다 빠른 탐색을 할 수 있는 알고리즘을 개발하고자 한다. 마지막으로 제안 방법에 기반하여 입력되는 스트림 데이터의 범위가 동적으로 변경되는 환경에서도 질의 색인 기법을 적용할 수 있는 스트림 데이터 시스템을 개발하고자 한다.

참 고 문 헌

[1] D.Carney, U. Cetintemel, M. Cherniack, C. Convey, S. Lee, G. Seidman, M. Stonebraker, N. Tatbul and S. Zdonik, “Monitoring stream – a new class of data management

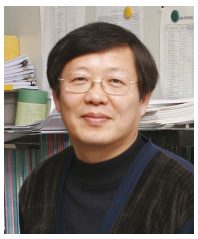
applications", *In Proc. of Very Lage Data Bases*, 2002.
[2] B. Babcock, S. Babu, M. Datar, R. Motwani, and J. Widom, "Models and Issues in Data Stream systems", *Proc. of ACM PODS 2002*, Madison, Wisconsin, United States, 2002.
[3] H. Edelsbrunner. "Dynamic data structures for othogonal intersection queries", Technical Report 59, institute for Information Processing, Technical University of Graz, Graz, Austria, 1980.
[4] E. Hanson, ISlist.tar: A tar file containing C++ source code for IS-lists, http://www-pub.cise.ufl.edu/~hanson/IS-lists/.
[5] A. Guttman, "R-trees: A dynamic index structure for spatial searching", *ACM Computing Surveys*, Vol.30, No.2, 1998.6
[6] E. Hanson, M, Chaabouni, C. Kim, and Y. Wang, "A Predicate Matching Algorithm for Database Rule Systems", In *Proc. of ACM SIGMOD 1990*, 1990.
[7] E. Hanson and T. Johnson, "Selection Predicate Indexing for Active Database Using Interval Skip Lists", Information Systems, Vol.21, No.3, 1996.
[8] S. R. Madden, M. A. Shah, J. M. Hellerstein, and V. Raman. "Continuously Adaptive Continuous Queries over Streams", *Proc. of ACM SIGMOD 2002*, Madison, Wisconsin, United States, 2002.
[9] H. Samet, *Design and Analysis of Spatial Data Structures*, Addison-Wesley, 1990.
[10] K.-L. Wu, S.-K. Chen, and P. S. Yu, "Interval query indexing for efficient stream processing", In *CIKM 2004*, pp.88-97, 2004.11.
[11] K.-L. Wu, S.-K. Chen, P. S. Yu and M. Mei. "Efficient interval indexing for content-based subscription e-commerce and e-serviec", In *Proc. of IEEE Int. Conf. on e-Commerce Technology for Dynamic E-Business*, 2004.12.
[12] Motwani, R. et al., "Query Processing, Approximation, and Resource Management in a Data Stream Management System", In *Proc. The First Biennial Conf. on Innovative Data Systems Research*, Asiloma, California, pp.245-256, 2003.1.
[13] Terry, D. et al., "Continuous Queries over Append-Only Databases", In *Proc, Int'l Conf. on Management of Data, ACM SIGMOD*, San Diego, California, pp.321-330, 1992.6.

김 재 인
e-mail : sereno3@naver.com
2008년 전남대학교 전자컴퓨터공학부(학사)
2008년~현 재 전남대학교 전자컴퓨터공학부 석사과정
관심분야 : 스트림 데이터, 연속질의, USN 응용

송 명 진
e-mail : audwls0324@nate.com
2009년 전남대학교 전자컴퓨터공학부(학사)
2009년~현 재 전남대학교 전자컴퓨터공학부 석사과정
관심분야 : 데이터 마이닝, 스트림 데이터, 알고리즘

한 대 영
e-mail : nara9yo@gmail.com
2008년 전남대학교 전자컴퓨터공학부(학사)
2008년~현 재 전남대학교 전자컴퓨터공학부 석사과정
관심분야 : 데이터 마이닝, 멀티미디어, 임베디드

김 대 인
e-mail : dikim@chonnam.ac.kr
1998년 전남대학교 전산통계학과(이학석사)
2006년 전남대학교 전산통계학과(이학박사)
2004년~현 재 전남대학교 전자컴퓨터공학부 시간강사
관심분야 : 스트림 데이터, 데이터 마이닝, 디지털 콘텐츠



황 부 현

e-mail : bhhwang@chonnam.ac.kr

1978년 숭실대학교 전산학과(학사)

1980년 한국과학기술원 전산학과(공학석사)

1994년 한국과학기술원 전산학과(공학박사)

1980년~현 재 전남대학교 전자컴퓨터공학부 교수

관심분야 : 스트림 데이터 마이닝, 분산 시스템, 분산 데이터베이스