비정형 빅데이터의 실시간 복합 이벤트 탐지를 위한 기법

The Method for Real-time Complex Event Detection of Unstructured Big data

이 준 희[*] 백 성 하[**] 이 순 조[***] 배 해 영[****]
Jun Heui Lee Sung Ha Baek Soon Jo Lee Hae Young Bae

요 약 최근 소셜 미디어의 발달과 스마트폰의 확산으로 SNS(Social Network Service)가 활성화가 되면서 데이터양이 폭발적으로 증가하였다. 이에 맞춰 빅데이터 개념이 새롭게 대두되었으며, 빅데이터를 활용하기 위한 많은 방안이 연구되고 있다. 여러 기업이 보유한 빅데이터의 가치창출을 극대화하기 위해 기존 데이터와의 융합이 필요하며, 물리적, 논리적 저장구조가 다른 이기종 데이터 소스를 통합하고 관리하기 위한 시스템이 필요하다. 빅데이터를 처리하기 위한 시스템인 맵리듀스는 분산처리를 활용하여 빠른게 데이터를 처리한다는 이점이 있으나 모든 키워드에 대해 시스템을 구축하여 저장 및 검색 등의 과정을 거치므로 실시간 처리에 어려움이 따른다. 또한, 이기종 데이터를 처리하는 구조가 없어 복합 이벤트를 처리하는데 추가 비용이 발생할 수 있다. 이를 해결하는 방안으로 기존에 연구된 복합 이벤트 처리 시스템을 활용하여 실시간 복합 이벤트 탐지를 위한 기법을 제안하고자 한다. 복합 이벤트 처리 시스템은 서로 다른 이기종 데이터 소스로부터 각각의 데이터들을 통합하고 이벤트들의 조합이 가능하며 스트림 데이터를 즉시 처리할 수 있어 실시간 처리에 유용하다. 그러나 SNS, 인터넷 기사 등 텍스트 기반의 비정형 데이터를 텍스트형으로 관리하고 있어 빅데이터에 대한 질의가 요청될 때마다 문자열 비교를 해야 하므로 성능저하가 발생할 여지가 있다. 따라서 복합 이벤트 처리 시스템에서 비정형 데이터를 관리하고 질의처리가 가능하도록 문자열의 논리적 스키마를 부여하고 데이터 통합 기능을 제안한다. 그리고 키워드 셋을 이용한 필터링 기능으로 문자열의 키워드를 정수형으로 변환함으로써 반복적인 비교 연산을 줄인다. 또한, 복합 이벤트 처리 시스템을 활용하면 인 메모리(In-memory)에서 실시간 스트림 데이터를 처리함으로써 디스크에 저장하고 불러들이는 시간을 줄여 성능 향상을 가져온다.

키워드 : 빅데이터, 복합 이벤트 처리 시스템, 복합 이벤트 탐지

Abstract Recently, due to the growth of social media and spread of smart-phone, the amount of data has considerably increased by full use of SNS (Social Network Service). According to it, the Big Data concept is come up and many researchers are seeking solutions to make the best use of big data. To maximize the creative value of the big data held by many companies, it is required to combine them with existing data. The physical and theoretical storage structures of data sources are so different that a system which can integrate and manage them is needed. In order to process big data, MapReduce is developed as a system which has advantages over processing data fast by distributed processing. However, it is difficult to construct and store a system for all key words. Due to the process of storage and search, it is to some extent difficult to do real-time processing. And it makes extra expenses to process complex event without structure of processing different data. In order to solve this problem, the existing Complex Event Processing System is supposed to be used. When it comes to complex event processing system, it gets data from different sources and combines them with each other to make it possible to do complex event processing that is useful for real-time processing specially in stream data. Nevertheless, unstructured data based on

[*] 인하대학교 컴퓨터정보공학과 공학석사과정 odllbo3@gmail.com
[**] 인하대학교 컴퓨터정보공학과 박사과정 bshzeratul@gmail.com
[***] 서원대학교 컴퓨터공학과 교수 sjlee@seowon.ac.kr(교신저자)
[****] 인하대학교 컴퓨터정보공학과 교수 hybae@inha.ac.kr

text of SNS and internet articles is managed as text type and there is a need to compare strings every time the query processing should be done. And it results in poor performance. Therefore, we try to make it possible to manage unstructured data and do query process fast in complex event processing system. And we extend the data complex function for giving theoretical schema of string. It is completed by changing the string key word into integer type with filtering which uses keyword set. In addition, by using the Complex Event Processing System and processing stream data at real-time of in-memory, we try to reduce the time of reading the query processing after it is stored in the disk.

Keywords : Bigdata, Complex Event Processing System, Complex Event Detection

1. 서 론

최근 IT 융합, 소셜 미디어, 기업들의 고객 데이터 수집활동, 멀티미디어 콘텐츠의 폭발적인 증가와 스마트폰 보급, SNS(Social Network Service)의 활성화로 전 세계에서 생산되는 데이터양이 활용 가능한 저장 용량을 초과하였으며, 앞으로도 기하급수적으로 증가할 것으로 예측된다. 이에 따라 빅데이터의 개념이 생겨나면서 최근 특정 분야에 국한되지 않는 가장 중요한 이슈로 다루어지고 있다. 특히 빅데이터는 기존의 고정된 필드에 저장하여 사용되는 정형 데이터 외에 고정되지 않은 비정형 데이터(예, 페이스북, 트위터 등의 텍스트 문서) 처리가 중요 하므로 비정형 데이터에 관한 분석연구가 활발히 진행되고 있다[7]. 더하여 여러 센서로부터 받은 이기종 데이터 소스를 토대로 소비자들의 이동 패턴이나 소비 패턴 등을 분석하여 기업이나 기관에서의 의사 결정 지원을 위한 정보 제공이 요구되고 있다[3, 6]. 따라서 빅데이터 분석뿐만 아니라 서로 다른 이기종 데이터 소스와 빅데이터의 융합이 추가로 요구된다.

맵리듀스(MapReduce)[1]는 구글에서 탄생한 프레임워크로 오늘날 대부분 빅데이터 처리에서 사용된다. 맵리듀스의 혁신적인 부분은 데이터 집합에 대한 질의를 입력받아 분할한 후, 여러 개의 노드에서 병렬로 처리하는 분산처리로서 단일 장비에서 처리하기 부적합한 대규모 데이터의 문제를 해결한다. 하지만 맵리듀스의 경우 빅데이터의 빠른 처리를 위한 키(key) 값을 이용한 데이터 통합 및 집합, 분산처리 및 저장 등에 초점이 맞춰져 있어 물리적, 논리적으로 상이한 이기종 데이터 소스에 대한 데이터 통합이나 스키마 매칭 및 통합 기능이 없으므로 복합 이벤트 처리에는 적합하지 않다. 예를 들어 스포츠 매장에서 축구에 관심 있는 고객들에게 축구용품에 대한 광고를 해주도록 요구할 수 있다. SNS를 통해 축구에 관련된 글을 올리는 사람들의 정보를 획득하여 글 작성자 근처의 해당 매장에서 축구용품에 관련된 광고를 보내주는 복합 이벤트가 발생한다. 이러한 과정을 수행하기 위해서는 시스템에서 비정형 데이터를 처리할 수 있어야 하며, 작성자의 위치 정보와 함께 매장 정보를 조인하는 질의가 필요하다. 따라서 사용자가 축구에 관련된 글을 탐색하는 이벤트와 SNS 작성자 및 매장의 위치 등 각각의 이벤트의 조합을 통해 복합 이벤트를 처리하는 과정과 광고를 바로 보내줄 수 있는 실시간 처리가 필요하다. 하지만 맵리듀스의 경우 복합 질의를 지원하지 않으므로 각각의 데이터나 이벤트들을 맵리듀스화 하여 디스크에 저장하고 각각의 디스크에서 다시 데이터를 읽어 들여 조인연산을 해야 하는 과정이 필요하다. 이에 따라 저장 공간이 낭비될 수 있고 이기종 소스별로 프로그래밍을 해야하는 추가적인 과정이 필요하다. 또한 하나의 텍스트에서 하나의 키 값만 추출이 가능하므로 하나 이상의 키 값이 필요한 경우 중복해서 연산해야 하는 단점이 있다.

이러한 문제점들을 개선하기 위해 스트림 데이터를 실시간으로 처리하고 이기종 데이터 소스로부터 복합 질의 수행을 지원하는 복합 이벤트 처리 시스템을 활용하고자 한다[2, 5, 9, 10]. 복합 이벤트 처리 시스템은 대량의 이벤트 스트림을 대상으로 하며, 필터링 등의 기능을 수행하여 기업에서 발생하는 복잡한 이벤트들을 탐지하고 관리하기 위해 필요한 시스템이다. 하지만 복합 이벤트 처리 시스템은 비정형 데이터에 대한 최적화가 되어있지 않아 질의를 수행할 경우 일반 텍스트로 입력 받아 처리해야 하는 단점이 있다. 텍스트의 질의 수행을 위해

서는 텍스트에 정보 요구가 발생 될 때마다 문자열 매칭이 발생하게 되어 비효율적이며, 상대적으로 데이터 크기가 큰 비정형 데이터를 저장하고 관리하기에는 많은 메모리가 필요하므로 빅데이터 환경에 적합하지 않다.

본 논문에서는 복합 이벤트 처리 시스템의 어댑터 자료구조를 추가하여 빅데이터의 텍스트형 비정형 데이터를 입력받아 질의 수행을 할 수 있도록 한다. 비정형 텍스트 데이터에 매핑되는 키워드 셋을 정의 하여 문자열의 키워드를 정수형으로 변환하고 비정형 데이터를 추상화함으로써 반복되는 문자열 매칭 연산을 줄인다. 또한, 필터링 기능을 이용하면 시스템의 사용자가 관심 있는 데이터를 선별함으로써 데이터를 분석하고 저장하는 시간을 줄여줄 수 있다.

본 논문의 구성은 다음과 같다. 2장에서는 관련 연구로 빅데이터, 맵리듀스와 복합 이벤트 처리 시스템 및 어댑터에 대해 기술하고, 3장에서는 이 논문에서 제안한 복합 이벤트 처리 시스템의 어댑터와 데이터 구조 및 자료구조, 질의예제에 대해 설명한다. 4장에서 실험을 통해 성능을 평가하고, 마지막 5장에서 결론을 맺고 향후 연구 방향에 대해 기술한다.

2. 관련 연구

2.1 맵리듀스(MapReduce)

웹 서치엔진 인덱싱 문제를 해결하기 위하여 구글에서 탄생한 맵리듀스[1] 프레임워크의 혁신적인 부분은 데이터 집합에 대한 질의를 입력 받아, 분할한 후, 여러 개의 노드에서 병렬로 처리하는데 그 핵심이 있다. 이러한 분산처리는 단일 장비에서 처리하기에는 부적합한 대규모 데이터의 문제를 해결한다. 그림 1은 맵리듀스의 구조를 나타낸다.

맵리듀스의 작업은 맵(Map)과 리듀스(Reduce)의 두 가지 단계로 나누어진다. 각 단계는 입력과 출력으로써 <키(key), 값(value)> 쌍을 가지고 있고, 그 타입은 사용자가 프로그래밍할 때 선택한다. 맵 단계에서는 입력된 데이터를 <키, 값>의 쌍으로 데이터를 정리하고 비정상적인 데이터는 배제시킨다. 키를 중심으로 <키, 값> 쌍들을 정렬하고 그룹을 만들어 임시 데이터 집합으로 변형 된다. 리듀스 단계에서는 맵 단계에서 생성된 <키, 값> 데이터들을 이용해서 처리하는 단계로 사용자의 정의에 따라 요약 혹은 구성을 통해 결과 값을 출력하며, 최종 결과물은 요약 데이터집합으로 환원된다. 위의 과정에서 개발자는 맵 함수와 리듀스 함수만을 구현하면 된다. 나머지 분산, 병렬 처리와 같은 과정은 맵리듀스 프레임워크에서 자동으로 처리해준다.

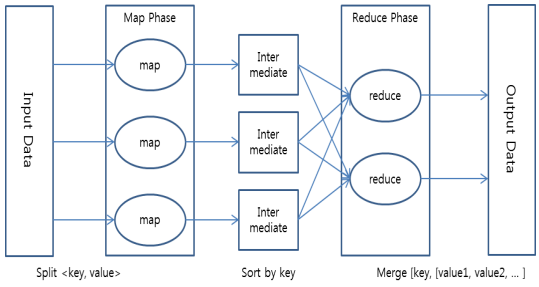

그림 1. 맵리듀스(MapReduce)의 구조

하지만 맵리듀스는 키와 값을 가지고 개별의 이벤트에 대한 처리는 빠르게 할 수 있으나 모든 문제에 적용할 수 없는 단점이 있다. 그리고 맵리듀스의 경우 서로 다른 이기종 데이터 소스에 대한 데이터 통합 및 스키마 통합의 기능이 없고 단순 키워드 검색을 지원한다. 또한, 서로 다른 데이터 소스로 부터 이벤트를 검출하기 위해선 소스별로 시스템을 구축해야하고 데이터를 디스크에 저장한 후, 디스크로부터 검색을 해야 하므로 실시간 수행 시 성능이 다소 떨어진다.

2.2 복합 이벤트 처리(CEP: Complex Event Processing)

2.2.1 u-GIS DSMS CEP 시스템

u-GIS DSMS는 유비쿼터스 시대를 위해 새롭게 요구되는 u-GIS 환경의 데이터 스트림 처리 시스템(DSMS: Data Stream Management System)와 지리 정보 시스템(GIS: Geography Information System)이 결합된 CEP 플랫폼이다. 유비쿼터스 환경을 기반으로 분산 편재된 GeoSensor로부터 발생하는 실시간 스트림 데이터에 대한 효과적인 저장과 실시간 연속 질의 처리를 통해 사용자의 상황에 부합하는 다양한 유비쿼터스 응용 서비스의 구축을 지원하는 GeoSensor 데이터 스트림 시스템으로 위치 및 이동성을 갖는 데이터, 멀티미디어, 모바일 등의 다양한 센서 데이터 스트림 환경을 기반으로

광역 수준의 센서 데이터를 통합하고 제어하여 위치 및 공간 정보 기반의 다양한 GeoSensor 데이터 스트림 처리를 통해 다양한 u-GIS 응용 서비스에 활용할 수 있는 기능을 제공한다[12, 13, 14, 15].

복합 이벤트 처리 시스템은 의미 있는 이벤트와 무의미한 이벤트가 같이 발생하는 대량의 이벤트 스트림을 대상으로 하며, 필터링 등을 수행하여 복합 이벤트 처리는 기업에서 발생하는 복잡한 이벤트들을 탐지하고 관리하기 위해 필요한 시스템이다. 복합 이벤트 처리의 목적 중 하나는 복잡한 비즈니스 환경에서 수 없이 발생하는 이벤트들에 대한 이해를 돕는 것이다. 복합 이벤트 처리를 통해서 특정 이벤트가 무엇에 의해서 발생했는지 알 수 있고, 그것을 바탕으로 대응하는 룰을 만들어 실행에 옮길 수 있다. 결국, 복합 이벤트 처리는 어떠한 동작이나 결과를 처리해주는 바탕을 제공해 준다.

일반적인 복합 이벤트 처리에서 이벤트는 단순 이벤트(simple event)와 복합 이벤트(complex event)로 구성된다. 단순 이벤트는 하나의 이벤트로 표현되며, 발생한 이벤트들은 모두 의미 있는 이벤트로 간주하고 각각의 이벤트 내용에 따라 액션을 수행한다. 복합 이벤트는 단순 이벤트의 조합으로 표현되며, 여러 이벤트 소스로부터 발생한 이벤트를 대상으로 이벤트들의 영향을 분석하여 대응되는 액션을 수행한다. 또한 복합 이벤트는 단순 이벤트의 연관성을 기초로 만들어지며, 의미 있는 데이터로 변환하여 응용 계층으로 전송한다. 복합 이벤트 언어는 DBMS 에서 사용하는 SQL과 비슷하다. 기존 상용화된 이벤트 처리 시스템으로는 Microsoft의 Coral8, Streambase, SYBASE 이 있다[2, 5, 9, 10].

2.2.2 u-GIS DSMS 어댑터

이기종 데이터 소스는 종류가 다양하므로 u-GIS DSMS에서 이기종 데이터를 획득 및 처리하기 위한 전처리 어댑터가 필요하다.

그림 2는 어댑터의 전체 구조를 나타낸다. 데이터 자원 관리자(Data Source Manager)는 이렇게 다양한 데이터 소스로부터 데이터를 획득하며, 관리자가 데이터 소스의 정보를 이용할 수 있도록 기능을 제공한다. 다양한 데이터 소스들로부터 획득된 데이터의 스키마 형태는 데이터 소스에 따라 서로 다르다. 따라서 획득된 데이터의 스키마를 분석하여 조건에 맞는 값을 찾거나 내부 시스템에서 사용할 수 있는지 확인하는 과정이 필요하다. 따라서 데이터 타입 매핑 테이블을 이용하여 외부 데이터 타입과 내부 데이터 타입의 호환여부를 비교하고 내부 시스템에서 활용할 수 있도록 스키마 변환이 이뤄져야 한다. 어댑터의 메모리 관리자는 다양한 데이터 소스로부터 입력되는 데이터의 획득과정에서 사용하거나, 획득된 데이터를 질의 처리하여 중간결과를 저장하는데 사용한다. 또한, 최종 처리된 결과를 응용 서비스에 제공하기 위해서도 메모리 공간이 요구된다[13].

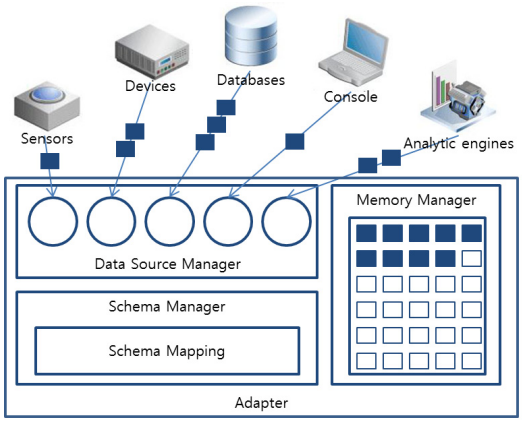

그림 2. 어댑터의 전체구조

2.2.3 u-GIS DSMS CEP의 한계

복합 이벤트 처리 시스템의 경우 스트림 데이터와 이기종 데이터 소스에 대한 복합 질의와 이벤트 검출 등의 다양한 역할을 수행하지만 빅데이터를 텍스트 기반으로 관리하고 있어 비정형 데이터에서 복합 이벤트를 검출하기 위한 시스템의 비용이 크다. 따라서 어댑터에 텍스트 기반의 비정형 빅데이터를 효율적인 데이터 처리를 위하여 데이터 구조와 자료구조를 추가하고 스키마를 부여하여 복합 이벤트 처리와 이벤트 검출 및 복합 이벤트 패턴 분석을 가능하게 한다. 추가로 키워드 셋을 이용한 필터링 기능을 추가하여 빅데이터를 추상화함으로써 이기종 데이터 소스들과의 질의를 가능토록 하며 사용자가 요구하는 필요한 정보들을 관리할 수 있도록 한다.

3. 본 문

기존의 복합 이벤트 처리 시스템의 언어를 확장하여 비정형 데이터를 입력받기 위해 어댑터의 데이터 구조를 추가하고 데이터 통합을 위해 자료구조를 정의한다. 비정형 데이터를 사용자의 관심사(Interesting)에 맞는 데이터를 추출하고 실시간 처리를 위한 키워드 셋을 이용한 필터링 기능에 대하여 설명한다. 마지막으로 비정형 데이터를 처리하는 과정의 예를 설명한다.

3.1 제안하는 복합 이벤트 처리 시스템의 구성

본 절에서는 빅데이터 처리를 위해 제안하는 복합 이벤트 처리 시스템의 전체적인 구성과 흐름을 설명한다.

그림 3은 제안하는 복합 이벤트 처리 시스템의 구조이다. 입력 데이터로서 기존의 정형(structured) 데이터 외에 비정형(unstructured) 데이터를 입력받을 수 있도록 어댑터에 접속정보에 관한 새로운 데이터 구조가 필요하다. 본 논문에서는 빅데이터의 비정형 데이터 중에서 SNS(Social Network Service)나 인터넷 뉴스 기사에서 주로 발생하는 텍스트 처리에 목적을 둔다.

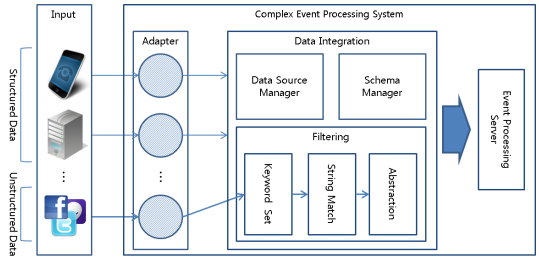

그림 3. 제안하는 복합 이벤트 처리 시스템의 구조

데이터 통합과정에서는 데이터와 스키마 관리를 위한 데이터 소스관리(Data Source Manager)와 스키마 관리(Schema Manager)기능을 수행한다. 또한 비정형 데이터를 처리하기 위해 새로운 필터링(Filtering) 기능을 추가한다. 필터링 기능은 크게 키워드 셋(Keyword set)을 이용하여 문자열 매칭을 통해 비정형 데이터를 추상화(Abstraction) 하는 것을 목적으로 한다. 키워드 셋은 사용자가 정의한 것으로 질의에 필요한 관심사(interesting)나 정보를 테이블 형태로 관리한다. 추상화는 키워드 셋을 이용하여 구조화 하는 것으로 텍스트 형태의 비정형 데이터에서 키워드가 발견되면 키워드를 정수형 형태로 변환을 하여 관리하고 원래의 텍스트(본문)는 링크정보만을 유지한다. 추상화의 목적은 크게 두 가지로, 하나는 비정형 데이터를 정형화에 가깝게 구조화를 해 놓음으로서 다른 이기종 데이터들의 데이터 통합을 통해 복합 이벤트 처리를 가능토록 한다. 비정형 형태의 텍스트에서 유용한 정보를 얻기 위해 문자열 매칭이 반드시 필요한데 문자열 매칭 알고리즘(String Matching Algorithm)의 경우 옵티말 타임이 최소 O(n+k)[1]로 상수시간에 해결되는 정수 비교 알고리즘에 비해 상당한 시간이 걸린다. 따라서 다른 하나는 질의에 필요한 문자열을 키워드 셋을 통해 정수형으로 변환 시켜 한번 변환이 된 문자열은 추후에 다시 문자열 매칭이 일어나지 않으며, 질의에 사용될 경우 비교 시간을 최소화하여 실시간 처리를 가능케 한다.

정형 데이터 및 비정형 데이터의 데이터 통합이 완료되면 이벤트 처리 서버로 보내어져 추가적인 이벤트 검출이나 복합 이벤트 같은 추가적인 처리가 가능하다.

3.2 질의처리를 위한 어댑터 자료구조

본 절에서는 텍스트형 비정형 데이터를 처리하는 어댑터의 기능을 수행하기 위한 자료구조에 대해 설명한다. 텍스트형 외의 동영상 같은 비정형 데이터의 경우 추가적인 자료 분석이나 데이터의 가공 및 수집 등이 필요하므로 본 논문에서는 복합 이벤트 처리에서 바로 유용하게 활용할 수 있는 텍스트형 비정형 데이터를 기준으로 한다.

SNS나 인터넷 뉴스 기사 등 텍스트 형태의 데이터를 가져오기 위해 어댑터에서 접속 정보에 관한 데이터 구조가 필요하다. 표 1은 비정형 데이터를 가져오기 위한 접속정보 관리구조이다. 페이스북, 트위터 등 SNS 종류(type)를 구분하기 위해 정수형 변수 “int”를 이용한다. 오픈API를 사용하기 위해서는 각 SNS로부터 발급받은 AppId가 필요하다. 따라서 오픈 API를 가져오기 위한 코드번호(appId)와

1) O(n+k): k는 문자열 매칭 알고리즘의 추가 연산 시간으로 알고리즘 마다 다르다. 예) The KMP(Kuth-Morris-Pratt) Algorithm 의 경우 Failure Function을 구축하는데 걸리는 시간

접속주소(url), 포트번호(port)는 가변길이 변수 “String”을 이용한다.

표 1. 비정형데이터 접속정보 관리 구조

```
public class UnsConnectInfo
{
  public int      id;     // 구분자
  public int      type;   // 종류구분(SNS 종류 등)
  public String   appId;  // 오픈API를 가져오기 위
                          한 AppID
  public String   url;    // 접속을 위한 주소
  public String   port;   // 접속을 위한 포트
}
```

비정형 데이터의 경우 복합 이벤트 처리 시스템에서 질의 처리가 어렵다. 비정형 데이터에서 유용한 정보를 얻어내기 위하여 문자열 비교가 필요하며 이를 정형화 하여 질의 처리와 이벤트 검출이 가능하도록 구조화된 형식으로 데이터 변환이 필요하다.

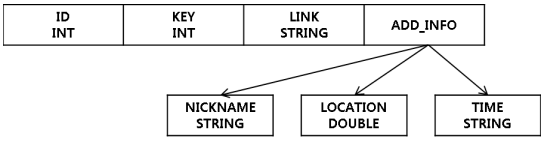

그림 4. 비정형 데이터를 위한 자료구조

그림 4는 비정형 데이터를 질의 처리 가능토록 변환한 자료구조이다. 각 데이터 별로 구분을 짓기 위한 기본적인 ID가 필요하며 키워드 셋을 통하여 키워드가 매칭이 발생할 경우 그에 해당하는 정수형 값을 저장해 놓을 공간이 필요하다(KEY). 비정형 데이터의 텍스트로 작성된 글이나 기사 등의 내용에 키워드와 매칭되는 정보가 하나 이상 있을 수 있으므로 정수형 변수 “int”형의 배열로서 관리한다. 본문의 내용을 모두 메모리나 디스크에 저장할 경우 비정형 데이터의 많은 정보량을 저장하기 어려우며 처리 속도 또한 현저히 떨어질 수 있다. 따라서 본문의 내용을 전부 저장하지 않고, 링크(LINK) 정보만 유지하여 효율적인 저장장치의 관리뿐만 아니라 저장으로 인한 속도저하를 줄일 수 있다. LINK의 경우 가변길이 "String"변수를 이용해 관리한다. 기본적으로 이러한 자료구조를 가지고 있으나 사용자에 의해 필요한 부가정보가 있을 수 있다. 따라서 동적으로 자료구조를 관리할 수 있도록 포인터를 가지고 있으며 프로그래밍에 의해 바뀔 수 있다. [그림4]에서는 추가적인 정보의 예로 SNS에 글을 작성한 사람의 닉네임(NICKNAME), 위치 정보(LOCATION), 작성시간(TIME)을 저장해 놓은 것을 확인할 수 있다.

표 2. 키워드셋(keyword set) 예

	키워드(keyword)	식별자(identifier)
1	축구	1102
2	농구	3625
3	배구	5423
4	야구	7726
5	수영	9912

표 2는 필터링(Filtering)을 위한 키워드 셋(keyword set)의 예로 테이블 형태로 관리된다. 필터링에서는 키워드(keyword)를 이용하여 문자열 매칭을 한다. 일치하는 키워드가 발생하면 해당 키워드의 식별자 값을 그림 4 자료구조의 KEY에 저장한다. 이 과정이 빅데이터에 대한 추상화 과정이며 필터링에서 제공하는 기능이다. 키워드 셋은 키워드와 그에 대한 식별자(identifier) 값으로 관리한다. 다만 키워드 셋에 정의 되어 있지 않은 문자열의 질의요구가 발생할 경우 맵리듀스와 마찬가지로 반복적인 문자열 매칭이 발생하게 되어 효율이 떨어질 수 있다. 따라서 시스템 사용자가 요구하는 이벤트에 알맞은 키워드 셋의 관리가 필요하다.

3.3 복합 질의 처리의 예

빅데이터 처리를 위해 제안하는 복합 이벤트 처리 시스템은 다음과 같은 역할을 수행한다. 예를 들어 복합 이벤트 처리 시스템을 사용하면 상업적인 용도의 광고에 유용하게 사용될 수 있다. 불특정 다수를 상대로 광고를 하는 것 보다 판매하려는 상품에 관심을 가지고 있는 사람들에게 광고를 해주는 것이 훨씬 큰 기대효과를 가져올 수 있으며 광고비용을 절감할 수 있을 뿐 아니라 광고비용 대비 효율도 높일 수 있다. 구체적으로 한 스포츠용품을 판매하는 회사에서 세일 중인 축구 물품을 축구에 관심 있는 고객들에게 광고를 해주는 경우를 들 수 있다.

그림 5는 빅데이터 질의 처리 예의 전체 구성을 나타낸다. 세 개의 이기종 데이터 소스로부터 데이터를 입력받아 복합 질의를 처리하는 과정에 대해

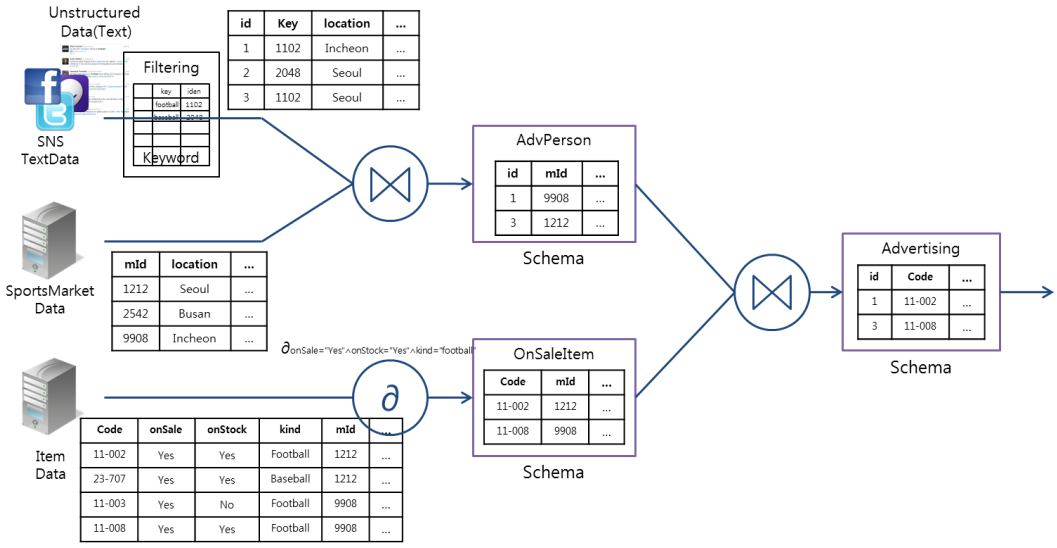

그림 5. 빅데이터 질의 처리 예의 전체 구성

설명한다. SNS TextData 에서는 텍스트형의 빅데이터를 발생한다. 비정형 형태로 들어온 데이터는 필터링의 표 2의 키워드셋을 이용하여 그림 4에서 언급한 자료구조로 변환된다. 변환되는 과정에서 문자열 형태의 키워드는 정수형으로 바뀐다. Sports Market Data에는 스포츠매장에 관한 정보들이 있으며, 각 매장의 고유마켓번호, 위치정보, 오픈시간, 전화번호, 주소 등의 정보를 관리하고 있다고 가정한다. Item Data에는 판매하는 품목들에 대한 정보를 가지고 있으며 상품번호, 상품명, 사이즈 및 치수, 가격, 세일중인지, 판매되는 고유마켓번호 등의 정보를 관리하고 있다고 가정한다. AdvPerson, OnSaleItem, Adverstising 은 스키마 로 XML형태의 질의가 필요하며 이벤트에 필요한 데이터의 중간과정을 저장하고 관리한다.

전체적인 과정은 먼저 SNS으로부터 입력받은 데이터에서 “축구”에 관련한 키워드를 가진 정보들을 추려내고 스포츠 매장정보와 조인연산을 통해 매장과 SNS 작성자의 위치가 5km 이내에 있는 소비자들을 가려내 AdvPerson 스키마에 저장한다. 상품정보가 저장된 Item Data에서는 할인중이고 재고가 남아있는 상품들을 선별해내어 OnSaleItem 에 저장한다. 마지막으로 AdvPerson과 OnSaleItem 으로부터 매장고유ID를 이용하여 조인연산을 통해 매장에서 판매중인 상품들을 선별해내는 질의처리 과정이다.

표 3. XML을 이용한 스키마 정의

스키마 1.

```
<CREATESCHEMA>
  <NAME>AdvPerson</NAME>
  <SQL>CREATE SCHEMA AdvPerson (sId INTEGER, sNickname VARCHAR(32), mId INTEGER, mTel VARCHAR(32), mLocation POINT)
  </SQL>
</CREATESCHEMA>
```

스키마 2.

```
<CREATESCHEMA>
  <NAME>OnSaleItem</NAME>
  <SQL>CREATE SCHEMA OnSaleItem (iId INTEGER, iName VARCHAR(32), iCode VARCHAR(32), iSize INTEGER, iPrice INTEGER, mId INTEGER)</SQL>
</CREATESCHEMA>
```

스키마 3.

```
<CREATESCHEMA>
  <NAME>Advertising</NAME>
  <SQL>CREATE SCHEMA Advertising (sId INTEGER, sNickname VARCHAR(32), mTel VARCHAR(32), mLocation POINT, iName VARCHAR(32), iSize INTEGER, iPrice INTEGER)</SQL>
</CREATESCHEMA>
```

예제와 같은 질의처리를 위해 XML을 이용하여 스키마를 정의하고 질의처리 하는 과정을 설명한다. 표 3은 XML을 이용한 스키마 정의를 나타낸다. 여기에서 사용된 XML은 기존에 [15]에서 연구된 XML을 기준으로 작성되었다. 스키마 1은 SNS와 SportsMarket 으로 부터 데이터를 입력받아 Adv Person에 중간 과정을 저장하기 위한 스키마다. 입력받은 비정형 데이터(SNS)로 부터 고유ID(sId), 닉네임(Nickname) 정보를 가져오고 SportsMarket 으로 부터 매장ID(mId), 전화번호(mTel), 매장위치(mLocation)의 정보를 유지한다. 스키마 2는 Item 으로 부터 입력을 받아 정보를 유지한다. 상품이름(iName), 상품코드(iCode), 상품사이즈(iSize), 가격(iPrice), 판매되는 판매처의 고유ID(mId)를 관리한다. 스키마 3은 중간 결과 값인 Intermediate, AdvPerson 과 OnSaleItem으로 부터 필요한 정보를 추출하여 Advertising에 필요한 정보를 저장한다. 광고를 하기 위한 목적이므로 빅데이터의 고유ID(sId), 작성자 닉네임(sNickname), 매장 전화번호(mTel), 매장 위치(mLocation), 상품명(iName), 사이즈(iSize), 가격(iPrice)의 정보를 저장하여 광고에 필요한 정보들을 저장한다.

표 4는 스키마를 이용하여 질의처리를 하기 위한 XML 언어이다. 질의 1은 스키마 1을 위한 질의처리다. INTO 절을 통해 AdvPerson 스키마에 정보를 저장하며 SELECT 절을 통해 튜플들이 정의 되어 있다. FROM 절에 의해서 SNS와 Sports MArket으로 부터 데이터 값을 가져오는 것을 알 수 있으며 WHERE 절에서는 키워드 “축구”에 관한 정보를 가져오고 작성자의 위치와 마켓의 위치를 확인하여 거리가 5km 이하인 작성자들의 정보를 가져온다. 작성자의 위치 정보를 고려한 것으로 소비자와 가까운 거리의 매장의 광고를 전해 주도록 한다. SNS.key = 1102 는 “축구”에 관한 키워드를 가져오는 것으로 표 2의 키워드셋을 보면 키워드 “축구”에 관한 식별자 값 “1102”를 확인할 수 있다. DISTANCE 는 거리에 관한 질의처리로 여기에서는 잠정적 소비자와 매장의 위치가 5km 이내일 경우에만 질의를 처리하게 한다. 질의 2는 스키마 2에 관한 질의처리로써 INTO 절을 통해 OnSaleItem 스키마에 저장한다. WHERE 절의 Item.onSale = ″Yes″ 는 할인중인 상품을 찾는 질의이며 Item. onStock = ″Yes″ 는 재고여부를 확인하는 질의이다. 따라서 현재 할인중이고 재고 있는 상품들을 찾는 질의이다. 마지막으로 질의 3은 스키마 3을 위한 질의로 INTO 절을 통해 Advertising 스키마에 저장하는 것을 확인할 수 있다. SELECT 문을 통해 광고에 필요한 여러 정보들을 가져온다. 스키마 1,

표 4. XML을 이용한 질의

질의 1.

```
<GEOSTREAMQUERY>
  <NAME>AdvPerson_Query</NAME>
  <GQQUERY>
    INTO AdvPerson
    SELECT SNS.sId, SNS.sNickname, Mar.mId,
           Mar.mTel, Mar.mLocation
    FROM SNS, SportsMarket AS Mar
    WHERE SNS.key = 1102 AND
          DISTANCE (SNS.pos, Mar.pos) <= 5000
  </GQQUERY>
</GEOSTREAMQUERY>
```

질의 2.

```
<GEOSTREAMQUERY>
  <NAME>OnSaleItem_Query</NAME>
  <GQQUERY>
    INTO OnSaleItem
    SELECT iId, iName, iSize, iPrice, mId
    FROM Item
    WHERE Item.onSale = ″Yes″ AND
          Item.onStock = ″Yes″ AND Item.kind
          = ″football″
  </GQQUERY>
</GEOSTREAMQUERY>
```

질의 3.

```
<GEOSTREAMQUERY>
  <NAME>Advertising_Query</NAME>
  <GQQUERY>
    INTO Advertising
    SELECT Per.sId, Per.sNickname, Per.mTel,
           Per.mLocation, Itm.iName, Itm.iSize,
           Itm.iPrice
    FROM AdvPerson AS Per, OnSaleItem AS Itm
    WHERE Per.mId = Itm.mId
</GQQUERY>
</GEOSTREAMQUERY>
```

2에 저장된 값들을 가져와 조인연산을 하며 WHERE 절은 스포츠매장의 고유ID와 상품이 판매되는 매장의 고유ID를 비교함으로 해당 매장에서 판매하는 상품들에 한해서만 보여줄 수 있도록 하는 질의이다.

4. 성능평가

본 장에서는 제안하는 복합 이벤트 처리 시스템의 빅데이터 처리의 성능을 평가한다. 성능평가는 맵리듀스와 제안하는 복합 이벤트 처리 시스템의 처리속도, 키워드 개수에 의한 성능을 비교한다. 이에 따른 실험 환경은 4.1절에서 설명하며 4.2절에서는 성능평가 결과를 설명한다.

4.1 평가환경

실험 평가에 사용된 시스템 환경은 CPU AMD Phenom IIX4 955 Processor 3.2 GHz, 메모리 4GB, 운영체제 Window 7 에서 시뮬레이션 하였다. Eclipse 개발 툴을 이용하여 Java jdk 1.7 환경에서 성능평가를 위해 텍스트형 비정형 데이터를 기준으로 테스트 모듈을 개발하였다. 또한, 성능평가에 이용된 데이터베이스로는 Oracle 10g를 이용하였다. 맵리듀스와 Linux 환경을 구성하기 위하여 Cygwin을 사용하였다. Cygwin은 Windows 환경에서 Linux 환경을 만들어 주는 프로그램이다. 맵리듀스를 위한 하둡의 버전은 Hadoop-0.20.2 버전을 사용하였다. 복합 이벤트 처리 시스템은 기존에 연구된 GSS (GeoSensor Data Stream processing System)[15]와 SYBASE[11]를 사용하였다.

4.2 성능평가

본 절에서는 제안 기법의 성능을 분석하기 위하여 4.1절에서 제시한 실험 환경에서 실험을 진행한다. 제안하는 CEP 시스템과 기존에 연구된 CEP 인 GSS 및 SYBASE를 사용하여 성능평가를 진행하였다. 이벤트가 발생하여 질의를 처리하는 과정에서 문자열 매칭이 필요하다. 질의 요청이 반복되어 발생할 경우 시스템 성능을 비교하기 위하여 질의 요구 회수에 따른 처리속도를 확인하였다. 시뮬레이션을 위해 2만개의 200 자 내외의 임의의 텍스트 데이터를 생성하여 사용했으며, 질의는 스포츠 관련(예, football, basketball, baseball, table tennis 등) 된 텍스트를 검색하는 것을 기초로 하였다.

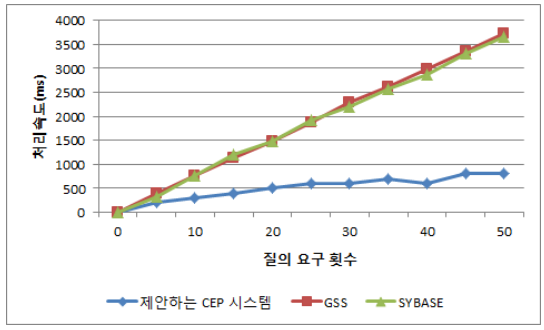

그림 6. 질의 요구 횟수에 따른 처리속도 비교

그림 6은 질의 요구 횟수에 따른 복합 이벤트 처리 시스템간의 처리속도를 비교한 것이다. 기존에 연구된 GSS 및 SYBASE 시스템의 경우 비슷한 성능을 가지며 제안하는 CEP 시스템의 경우 다른 두 시스템에 비해 처리속도가 빠른 것을 확인할 수 있다. 기존의 CEP 시스템의 경우 질의 요청이 발생할 때마다 텍스트를 읽어 문자열 매칭 연산을 해야 하므로 비효율적이다. 반면에, 제안하는 CEP 시스템의 경우 한번 텍스트를 읽어 들여 키워드 값에 대해 정수형으로 변환하여 정형화된 데이터 구조로 변환을 하면, 한번 문자열 매칭이 이루어진 이후에는 정수형으로 비교를 할 수 있기 때문에 처리속도 면에서 성능향상을 기대할 수 있다.

텍스트형 비정형 데이터를 맵리듀스 대신 CEP 시스템을 이용함으로써 인메모리(In-memory)에서 질의처리가 가능하다. 이에 따라 맵리듀스 과정에서 필요한 저장장치의 입·출력 시간을 단축할 수 있다. 이에 따른 성능평가로 그림 7은 인메모리에서 즉시 질의처리가 가능한 제안하는 CEP 시스템과 키/값을 이용하여 디스크에 저장한 후 값을 읽어 질의를 처리해야 하는 맵리듀스간의 성능분석을 위한 비교이다. 맵리듀스는 태스크를 하나만 사용하여 분산처리 기능은 사용하지 않았다. 텍스트 형의 레코드 수를 증가시키며 처리속도를 비교하였다. 레코드 수가 증가함에 따라 제안하는 CEP 시스템이 맵리듀스에 비해 처리속도가 빠른것을 확인할 수가 있으며, 레코드 및 데이터양이 증가할 수록 그 차이는 더 많이 발생할 것으로 예상된다. 맵리듀스를 이용할 경우 스트림 데이터에 대한 처리과정이 없으므로 질의를 처리하기 위해선 맵·리듀스 단계를 거쳐 데이

터를 축소화하여 디스크에 저장하고 다시 읽는 과정을 거쳐야 하므로 디스크의 입·출력 시간만큼 처리속도가 오래 걸린다.

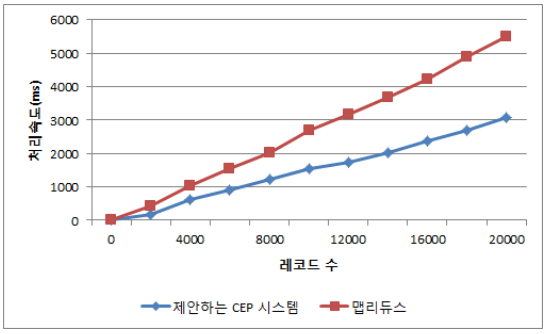

그림 7. 레코드 수에 따른 처리속도 비교

5. 결론 및 향후연구

본 논문에서는 빅데이터와 이기종 데이터의 질의처리를 위한 방안을 연구하였다. 맵리듀스는 스트림 데이터 및 복합 질의를 처리하는데 비효율적이며 기존의 복합 이벤트 처리 시스템은 빅데이터를 처리하는데 비효율적인 문제가 있다. 이를 개선하기 위하여 기존의 복합 이벤트 처리 시스템에 어댑터 부분을 확장하고, 이기종 데이터와 빅데이터의 복합 질의 처리를 위하여 데이터 통합을 하고자 새로운 자료구조를 정의하였다. 필터링 기능을 사용하여 문자형의 키워드 값을 정수형으로 변환하여 질의 수행에 필요한 문자열 매칭의 횟수를 줄여 속도향상을 가져왔다. 비정형 빅데이터에 대한 데이터들을 디스크에 모두 저장하지 않고 데이터 구조로 추상화하여 디스크 비용을 줄이고 디스크 입·출력 시간을 줄이고자 하였다.

성능평가에서는 이벤트에 대한 질의 요청시 문자열 매칭이 필요한데 문자열 매칭 횟수에 대한 시스템간 성능을 비교하기 위하여 질의 요구 횟수에 따른 처리속도를 확인하였다. 확인결과 제안하는 CEP 시스템은 반복적으로 질의 요청이 발생하여도 문자열 매칭은 한번만 이루어지므로 다른 복합 이벤트 처리 시스템에 비해 처리속도가 빠른 것을 확인하였다. 또한 인메모리(In-memory)에서 스트림 데이터 형태의 질의처리가 가능한 복합 이벤트 시스템의 성능을 확인하기 위하여 맵리듀스와 처리 속도를 비교하였다. 디스크에 대한 입·출력이 발생하지 않아 제안하는 CEP 시스템이 맵리듀스에 비해 속도가 향상된 것을 확인하였다. 이 연구는 개인 데이터를 기반으로 사용자의 행동 패턴, 선호등을 분석하여 마케팅 및 서비스를 제공할 수 있다.

향후 연구로는 본 논문에서는 비정형 데이터 중 텍스트 형태의 빅데이터만 다루고 있어 그 외의 다양한 형태의 빅데이터를 처리할 수 있는 연구가 필요하다. 또한, 키워드셋의 갯수에 따라 시스템 성능 저하를 가져오므로 효율적으로 관리할 수 있는 연구가 필요하다.

참 고 문 헌

[1] J. Dean, S. Ghemawat, 2008, "MapReduce; Simplified Data Processing on Large Clusters", Communications of the ACM, vol. 51, no 1, pp.107-113.

[2] Y. Diao, Neil Immerman, Daniel Gyllstrom, 2007, "SASE+: An Agile Language for Kleene Closure over Event Streams," In UMass Technical Report 07-03.

[3] B. Gedik, L. Liu, 2004, "ModiEyes: Distributed processing of continously moving queries on moving objects in a mobile system," Adavances in Database Techonology, vol. 2992, pp67-87.

[4] S. Ghemawat, H. Gobioff, S. Leung. 2003, "The Google file system," In Proc of ACM Symposium on Operating Systems Principles, Lake George, NY, Oct, pp29-43.

[5] D. Gyllstroml, E. Wu, H. Chae, Y. Diao, P. Stahlberg, G. Anderson, 2007, "SASE: Complex Event Processing over Streams," In CIDR' 07, Asilomar, CA, USA.

[6] H. Hu, J. Xu and D.L. Lee, 2005, "A generic framework for monitoring continuous spatial queries over moving objects," Proc. of the ACM SIGMOD International Conference on Management of Data, pp. 479-490.

[7] McKinsey, 2011, "Big Data: The Next Frontier for Innovation, Competition, and Productivity", [Online] McKinsey & Compnay, http:// www.mckinsey.com/.

[8] Apache Hadoop, http://hadoop.apache.org/, 2012

[9] "Complex Event Processing with Coral8 Final," 2009, http://www.microsoft.com/.
[10] "StreamBase Pattern Matching language," 2009, StreamBase, http://www.streambase.com/
[11] SYBASE, http://infocenter.sybase.com/, 2012.
[12] 강홍구, 박치민, 홍동숙, 한기준, 2007, "공간 센터 데이타의 효율적인 실시간 처리를 위한 공간 DSMS의 개발," 한국공간정보시스템학회지, 제9권, 제2호, pp.45-57.
[13] 신재완, 2010, "u-GIS DSMS에서 이기종 데이터 처리를 위한 어댑터 설계 및 구현," 인하대학교 대학원.
[14] 박치민, 홍동숙, 박춘걸, 한기준, 2006, "STREAM을 기반으로 하는 공간 DSMS의 설계 및 구현," 한국공간정보시스템학회 추계학술대회 U-방재 국토의 구현, pp.131-136.
[15] 정원일, 신승선, 백성하, 이연, 이동욱, 김경배, 이충호, 김주완, 배해영, 2009, "u-GIS 컴퓨팅을 위한 GeoSensor 데이터 스트림 처리 시스템," 한국공간정보시스템학회지, 제11권, 제1호, pp.9-16.

논문접수 : 2012.08.06
수 정 일 : 1차 2012.10.15 / 2차 2012.10.25
심사완료 : 2012.10.30

이 준 희

2007년 서원대학교 컴퓨터 교육과 이학사
2012년 인하대학교 컴퓨터정보공학 컴퓨터정보공학과 공학석사과정
관심분야는 빅데이터, 데이터베이스

백 성 하

2005년 인하대학교 수학과 이학사
2007년 인하대학교 컴퓨터공학부 공학석사
2007년~현재 인하대학교 정보공학과 박사과정
관심분야는 데이터 스트림 관리 시스템, 데이터베이스 클러스터, 위치기반서비스

이 순 조

1985년 인하대학교 전자계산학과 이학사
1987년 인하대학교 전자계산학과 이학사
1995년 인하대학교 전자계산학과 공학박사
1995년~1997년 대림대 전자 계산과 교수
1997년~현재 서원대학교 컴퓨터공학과 교수
관심분야는 데이터베이스, 실시간 데이터베이스 시스템, GIS, 데이터베이스 시스템의 보안

배 해 영

인하대학교 공학사(응용물리학), 연세대학 공학석사(전자계산학), 숭실대학교 공학박사(전자계산학), 인하대학교 정보통신대학원장, 대학원장 역임.
현 인하대학교 컴퓨터정보공학과 교수
지능형 GIS연구센터 센터장, 중국 중경우전대학교 명예교수
관심분야는 데이터베이스, 공간 데이터베이스