디지털산업정보학회 논문지 제13권 제2호-2017년 6월
http://dx.doi.org/10.17662/ksdim.2017.13.2.077

실내 환경 모니터링을 위한 빅데이터 클러스터 설계 및 구현*

전 병 찬** · 고 민 구***

Design and Implementation of Big Data Cluster for Indoor Environment Monitering

Jeon Byoungchan · Go Mingu

〈Abstract〉

Due to the expansion of accommodation space caused by increase of population along with lifestyle changes, most of people spend their time indoor except for the travel time. Because of this, environmental change of indoor is very important, and it affects people's health and economy in resources. But, most of people don't acknowledge the importance of indoor environment. Thus, monitoring system for sustaining and managing indoor environment systematically is needed, and big data clusters should be used in order to save and manage numerous sensor data collected from many spaces. In this paper, we design a big data cluster for the indoor environment monitoring in order to store the sensor data and monitor unit of the huge building Implementation design big data cluster-based system for the analysis, and a distributed file system and building a Hadoop, HBase for big data processing. Also, various sensor data is saved for collection, and effective indoor environment management and health enhancement through monitoring is expected.

Key Words : Big Data Cluster, Sensor Data, Hadoop, Monitoring, HBase

Ⅰ. 서론

대부분의 현대인들은 연령에 관계없이 실내에서 많은 시간을 보내는 사람들이 늘어나고 있다. 학생들은 학교나 학원에서, 직장인들은 회사에서 대부분의 시간을 보내게 되는데, 이들에게 있어 좋은 실내 환경의 질은 건물 내에서 기본적으로 갖추어야 할 요건이다. 또한 좋은 실내 환경의 질과 더불어 에너지 관리도 중요하다. 특히, 강의실, 사무실 등은 학생들이 학업에 매진하고, 직장인들이 업무에 매진하기 위해 많은 시간을 보내는 장소이기에 쾌적함을 더욱 유지해야하며, 사람이 없는 빈 공간에 대해서는 에너지 절약이 되어야 한다. 또한 실내 공기는 자연 희석률이 부족하여 특별한 공기 정화 장치가 없는 한 오염된 공기가 계속적으로 순환하여 실내 공기 질은 악화된다. 세계 보건기구는 실내 공기 질에 영향을 미치는 오염물질로 일산화탄소(CO), 이산화탄소(CO2), 미세먼지, 이산화질소(NO2), 중금속 등이 있으며, 실내공기오염의 원인으로는 환기부

* 본 연구는 2017년도 청운대학교 학술연구조성비 지원에 의하여 연구되었음.
** 청운대학교 기초교양교육원 교수 (주저자)
*** 순천향대학교 전산학과 (교신저자)

족, 실내흡연, 오염된 외부공기의 실내 유입 등이 있다. 그러나 실내 환경에 대한 중요성과 에너지 낭비를 사람들은 인지하지 못하고 있으며, 필요성을 느끼지 못하고 있다. 그러므로 이러한 사람들에게 실내 환경에 대한 상태를 알리고, 인식하도록 하는 것이 필요하다. 이에 따라 실제 실내 환경을 측정하고 개선하기 위해 각 실내 공간마다 Co2, 온도, 습도, 먼지, 조도 등의 센서를 설치하여 이들로부터 데이터를 수집 및 저장하고 모니터링과 분석이 가능하도록 한다. 이러한 환경에서 센서 데이터를 계속적으로 수집하고 모니터링 가능한 공간 단위가 확장된다면 데이터양은 기학적으로 증가하게 되며, 단일 서버의 관계형 데이터베이스에서는 거대한 데이터를 저장하고 분석하는 데에는 한계를 나타내게 된다.

본 논문에서는 실내 환경 개선과 관계형 데이터베이스가 가진 빅데이터 처리의 한계를 극복하고자 실내 환경 모니터링을 위한 빅데이터 클러스터 시스템을 제안한다. 여러 공간에서 수집되는 많은 양의 센서 데이터를 저장하고 모니터링하기 위해 하둡 분산 파일 시스템과 Hbase를 설치한 빅데이터 클러스터 서버를 구축하였다.

Ⅱ. 이론적 배경

2.1 실내 환경 모니터링

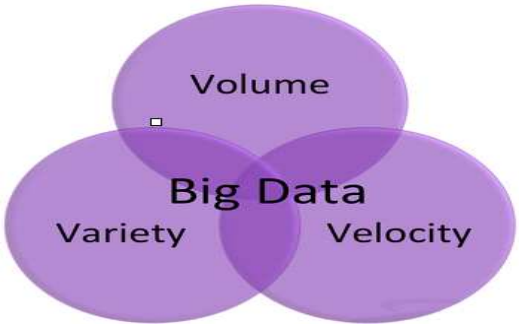
실내 환경 모니터링이란 실내 온도, 공기 질, 습도 등의 실내 환경에 대한 정보를 데이터화 하여 모니터링 하는 것을 말한다. 인구수의 증가로 주거 공간이 늘어났으며, 실내 업무를 하는 직장인들의 수도 증가 하였다. 또한 대부분의 놀이 문화가 실외보다는 실내에서 하는 것들이 늘어나면서 현대인들은 이동 시간을 제외한 대부분의 시간을 실내에서 보낸다[1-3].

<그림 1> 현대인의 1일 평균 실내에서 보내는 시간

그러나 실내에서 많은 시간을 보내는 만큼 실내 위험 요소에 노출되어 있지만, 사람들의 주변 실내 환경 의식은 매우 떨어진다. 특히 청소년과 아이들이 즐겨 찾는 PC방, 영화관 등의 놀이 시설은 폐쇄된 공간으로 공기 질이 매우 좋지 않음에도 불구하고 이용객들은 이를 인지하지 못하고 있으며, 업주들도 방관하고 인지하지 못하고 있다. 실제 한 매체의 조사에 따르면 영화관, PC방, 학원 등의 20% 가량이 실내 공기 오염도 기준치를 넘어섰다고 보도하고 있다. 그러나 정부는 이에 대한 대응책으로 1년에 한 번 실내 공기 질을 자체적으로 측정하고 보고하는 대응 방안을 내놓았으나 관리자가 적극적인 관리 의지를 보이지 않으면 부실한 대응책이다. 심지어 PC방의 경우 공기 질 개선을 위해 금연 정책까지 도입했으나 공기 질이 개선되지 않았다는 결과가 보고되었다. 실내 환경 모니터링은 이러한 문제들을 해소할 수 있는 기반이 될 수 있다. 다수의 넓은 공간을 인력을 통해 직접 점검하고 관리하기 보다는 실내 환경 모니터링을 도입하여 상시 모니터링하고 환기 설비를 갖춘다면 훨씬 효율적인 대응책이 될 수 있고 실내 환경의 쾌적함을 향상시킬 수 있다.

2.2 빅데이터

스마트 폰과 SNS(소셜 네트워크 서비스)의 등장으로 개인정보, 위치정보, 메시지, 이미지 등의 콘텐츠 정보가 급격히 증가하여 발생한 거대한 크기의 대용량 데이터를 '빅데이터' 라고 부른다. 이외에도 GPS를 기반으로 한 지도 정보, 날씨 정보 등 현존하는 모든 정보가 데이터라는 개념으로 정의되면서 빅데이터 시대가 등장했다. 그러나 무조건 데이터의 양이 거대하다해서 빅데이터라고 할 수 없다. 빅데이터란 양적, 질적으로 과거의 기술로는 정보로서 의미화가 불가능한, 기존 상식에서는 상상조차 할 수 없는 거대한 데이터를 지칭한다. 즉 일반적인 데이터베이스로는 관리하기 어려운 정도의 큰 규모의 대용량 데이터를 의미하며, 최근에는 대용량 데이터의 수집, 저장, 분석 등을 포괄하는 용어로 불리고 있다[4-6]. 빅데이터 기술은 빅데이터의 속성을 정확히 파악하고 적용하여야만 유용한 정보를 추출할 수 있다. <그림 2>에서 보여주는 바와 같이 그 속성은 데이터의 크기(Volume), 정형, 비정형 데이터와 같은 데이터의 다양성(Variety), 데이터 저장 및 처리속도(Velocity) 라는 세 가지를 꼽을 수 있으며 이를 '3V'라고 한다.



<그림 2> 빅데이터의 3V

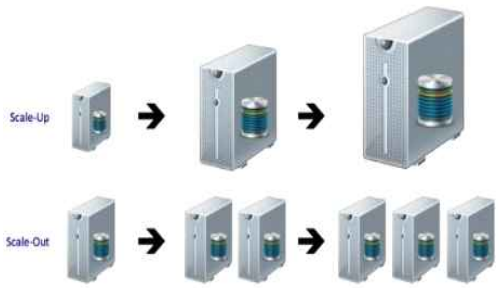
이러한 특징을 갖는 빅데이터는 다수의 데이터가 비정형 데이터로 이루어져 있으며, 이를 기술적으로 구조화 시켜서 유용한 정보를 추출하기 위해서는 새로운 형태의 처리 방식이 필요하다. 이에 따른 빅데이터와 같은 정형, 비정형 데이터를 처리하기 위한 인프라 기술 중 가장 대표적인 관심을 받는 기술로는 하둡이 있다.

Ⅲ. 빅데이터 클러스터 구축

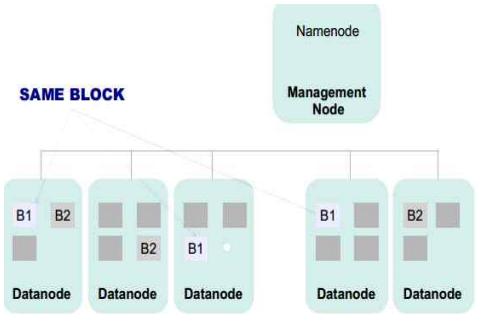
3.1 서버 구성도

단일 서버에서 서버의 스펙은 상당히 중요하다. 1대의 서버로만 구성 되어 있기 때문에 처리 속도, 성능저하를 막기 위해서는 서버 자체의 스펙이 높아야만 한다. 하지만 빅데이터를 처리하는데 있어서 단일 서버는 몇 가지 한계점을 보여 준다. 첫째로, 처리할 데이터의 용량이 커질수록 처리 시간이 오래 걸린다. 하드디스크 드라이브의 용량은 수년 동안 엄청나게 증가하여, 현재는 테라바이트 수준에 달하고 있다. 반면에 접근 속도는 약 100MB/s 수준에 머물러 있기에 1테라바이트의 디스크를 모두 읽어내는 데에는 두 시간 반 이상이 걸린다. 그러나 여러 대의 서버로 구성된 클러스터는 같은 용량의 데이터를 나누어 저장했기 때문에 병렬 처리를 하게 되면 처리 시간은 더 단축된다. 둘째, 빅데이터 처리를 위한 데이터베이스 성능은 단일 서버의 스펙 향상으로는 성능 향상을 기대하기 어렵다. 스펙 향상을 위해 들어가는 비용이 기하급수적으로 증가하기 때문이다. 이와 같은 방식을 스케일 업이라고 하며, 반대로 스케일 아웃은 적은 비용으로 클러스터에 새로운 서버를 추가함으로써 성능을 향상시킬 수 있고, 디스크 용량 확장성이 좋다. 마지막으로 데이터 유실 방지가 어렵다. 단일 서버에서 데이터 처리 혹은 저장 시에 데이터가 유실되거나 손실되면, 디스크

가 백업되어 있지 않으면 데이터 복원 가능성이 낮다. 하지만 클러스터에 설치된 하둡은 하나의 블록에 대해 3개의 복제 수준을 가지기 때문에 다른 노드(서버)의 블록이 유실되더라도 복원이 가능하다.

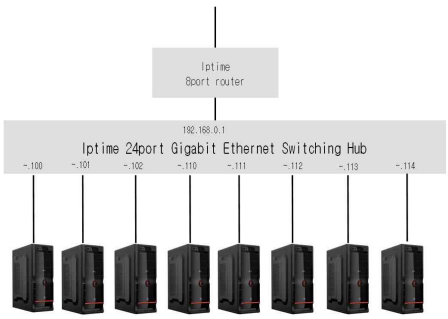


<그림 3> 스케일 업과 스케일 아웃



<그림 4> 하둡의 복제수준

<그림 5>는 본 논문에서 제안한 빅데이터 클러스터에 대한 구성도이다.



<그림 5> 빅데이터 클러스터 구성

빅데이터 클러스터 내의 각 노드들의 하드웨어 스펙 결정은 노드들이 맡게 될 역할에 따라 달라진다. 우선 하둡과 HBase에서의 소프트웨어적인 역할이 각 노드들에 배분되어야 한다. 8대의 노드는 모두 Datanode이며, 이 중에 2대는 각각 Namenode와 SecondaryNamenode의 역할을 수행한다. 각 노드들에는 고정된 IP를 할당하고, 2개의 호스트 이름을 지정한다. 호스트 이름은 각각의 노드들을 구분하기 위한 용도와 노드들의 역할을 구분하기 위한 두 가지 용도로 사용된다. 본 논문에서 사용되는 빅데이터 클러스터의 소프트웨어적인 역할은 다음 <표 1>과 같다.

<표 1> 빅데이터 클러스터 내의 각 노드들의 소프트웨어적 역할

번호	Hostname2	OS	CPU	RAM	HDD
1	master	Ubuntu 14.04.1 LTS	Intel® Core™ i3-4150 Processor	8GB	2TB
2	slave0	Ubuntu 14.04.1 LTS	Intel® Core™ i3-4150 Processor	8GB	2TB
3	slave1	Ubuntu 14.04.1 LTS	Intel® Core™ i3-4150 Processor	8GB	2TB
4	slave2	Ubuntu 14.04.1 LTS	Intel® Core™ i3-4150 Processor	8GB	2TB
5	slave3	Ubuntu 14.04.1 LTS	Intel® Core™ i3-4150 Processor	8GB	2TB
6	slave4	Ubuntu 14.04.1 LTS	Intel® Core™ i3-4150 Processor	8GB	2TB
7	slave5	Ubuntu 14.04.1 LTS	Intel® Core™ i3-4150 Processor	8GB	2TB
8	slave6	Ubuntu 14.04.1 LTS	Intel® Core™ i3-4150 Processor	8GB	2TB

빅데이터 클러스터에서 소프트웨어 역할 중 Namenode는 메모리 공간에 대한 스펙이 좋아야 한다. 다른 노드들을 관리하고 처리하는 작업이 수행되는 노드이기에 디스크 공간은 크지 않더라도 큰 메모리 공간이 필요하다. Datanode의 경우에는 데이터가 저장되는 노드이다. 따라서 빅데이터의 저장을 위해서 큰 디스크 공간을 필요로 한다. 그러나 본 논문에서는 기존 자원의 활용에 중점을 두었기 때문에 최적의 하드웨어 스펙을 맞추지는 않았다. <표 2>는 빅데이터 클러스터 내의 각 노드들의 하드웨어 스펙을 정리한 내용이다.

<표 2> 빅데이터 클러스터 내의 각 노드들의 하드웨어 스펙

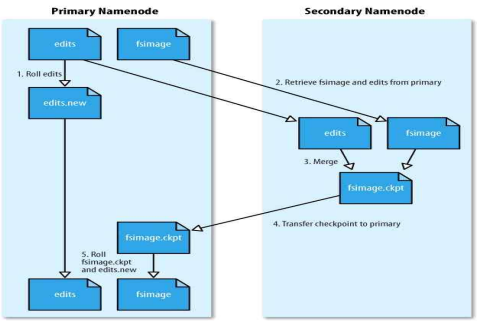
번호	Hostname	Hostname2	IP	Hadoop	Hbase
1	MINGOO	master	192.168.0.100	Namenode, Datanode	HMaster, HRegionServer
2	JUNGHYEN	slave0	192.168.0.101	Secondary Namenode, Datanode	HRegionServer
3	YESEUL	slave1	192.168.0.102	Datanode	HRegionServer
4	JINSIL	slave2	192.168.0.110	Datanode	HRegionServer
5	HEEJIN	slave3	192.168.0.111	Datanode	HRegionServer
6	JEONGJU	slave4	192.168.0.112	Datanode	HRegionServer
7	MUNSUNG	slave5	192.168.0.113	Datanode	HRegionServer
8	JINWOO	slave6	192.168.0.114	Datanode	HRegionServer

3.2 하둡 분산 파일 시스템

하나의 데이터 셋이 물리적인 서버의 스토리지 용량을 초과할 때 독립된 여러 서버에 전체 데이터 셋을 파티션 하여 저장할 필요가 있다. 네트워크로 연결된 서버의 스토리지를 관리하는 파일 시스템을 분산 파일 시스템이라 한다. 하둡은 HDFS(Hadoop Distributed FileSystem)이라는 분산 파일 시스템을 제공한다. 분산 파일 시스템의 가장 큰 도전 과제 중 하나는 노드 장애가 발생하더라도 데이터 손실이 발생하지 않도록 하는 것인데, 하둡은 전반적으로 노드 장애가 발생할 확률이 높은 범용 하드웨어의 클러스터에서 노드 장애가 발생하여도 중단 없이 작업을 수행 할 수 있도록 설계되었다[7-9]. 또한 네트워크에 연결된 스토리지를 이용해 수백~수천대 이상 확장 가능한 구조이며, 비교적 저렴한 비용으로 페타 바이트 이상의 데이터를 저장 할 수 있도록 설계 되었다. 단일 디스크를 위한 파일시스템처럼 HDFS 파일은 블록 크기의 청크(Chunk, 64MB)로 쪼개지고 독립적인 단위로 저장된다. 한 개의 블록에 대해서 여러 개의 복제본이 생성되며, 복제 수준은 최소 3개 이상을 가진다. 복제 된 블록들은 각 노드에 분산되어 저장된다. 이는 하드웨어의 결함을 보완하고, 데이터를 분산하여 높은 데이터 처리량을 제공하기 위해서 적용된 개념이다. 각 블록이 저장된 노드들의 위치와 정보는 파일 네임스페이스에서 관리하게 된다.

3.2.1 네임노드와 데이터 노드

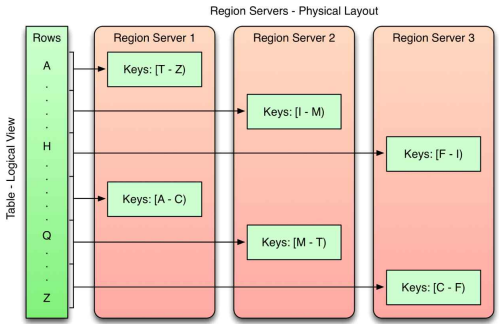
HDFS는 일반적인 분산 파일 시스템의 설계를 따라 마스터(master)노드와 슬레이브(slave)노드로 구성되어 있다. 마스터노드는 네임노드(namenode)라 불리며, 슬레이브 노드는 데이터노드(datanode)로 불리는데, 네임노드는 데이터 노드의 동작 상태를 실시간으로 관리하고 최대 수천 대의 데이터 노드에 분산 저장되어 있는 데이터에 대한 메타데이터를 관리하는 일을 한다. 각 블록이 저장된 서버의 위치와 정보는 네임노드에서 관리하지만, 위치 정보는 시스템 시작과 함께 데이터노드로부터 재구성되기 때문에 네임노드가 블록의 위치 정보를 지속적으로 저장하지는 않는다. 데이터노드는 클라이언트나 네임노드의 요청이 있을 때 블록을 저장하고 탐색하고, 주기적으로 저장하고 있는 블록의 리스트를 네임노드로 보고한다. 네임노드를 실행 중인 장비가 손상되면 데이터노드에 저장된 모든 블록을 재구성할 수 없게 된다. 이러한 이유로, 네임노드는 장애복구 능력을 갖추기 위해 보조(Secondary) 네임노드를 운영한다. 네임노드가 재구성 될 때 에디트 로그와 파일 시스템 이미지를 병합하는데, 이 때 에디트 로그가 지나치게 크다면 메모리에 로딩되지 못하거나 지연되어 네임노드가 구동되지 못하는 상황이 발생할 수도 있다. 이를 위해, 보조 네임노드는 에디트 로그가 너무 커지지 않도록 주기적으로 네임스페이스 이미지를 에디트 로그와 병합하고, 네임노드의 인메모리 메타데이터에 체크포인트를 생성한다. 체크포인팅 절차는 다음 <그림 6>와 같다.



<그림 6> 체크 포인팅 절차

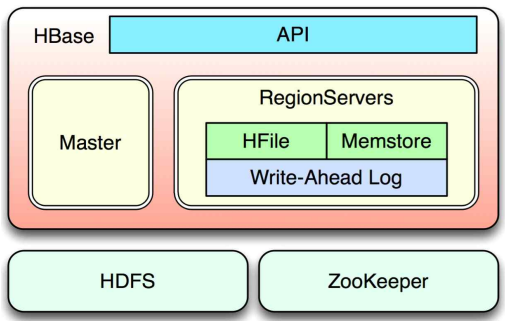
3.3. HBase

HBase는 HDFS에 구현한 분산 컬럼 지향(Distributed Column-Oriented) 데이터베이스로 대규모 데이터 셋에 실시간으로 랜덤 액세스가 필요할 때 사용할 수 있는 하둡 응용프로그램이다. HBase는 확장성을 위해 단지 노드만 추가하면 선형적으로 확장될 수 있는 구조를 제공한다. 응용 프로그램은 데이터를 테이블에 저장하며, 테이블은 로우와 컬럼으로 만들어진다[10-12]. HBase 테이블에서 가장 기본적인 단위는 컬럼이다. 하나 이상의 컬럼이 모이면 하나의 로우를 이루며, 로우는 로우키(Row-key)라는 유일한 주소를 가진다. 로우를 이루고 있는 컬럼은 차례대로 컬럼 패밀리로 그룹화 된다. 컬럼 패밀리는 데이터를 의미적으로 또는 주제별로 분류하게 해주며, 테이블이 생성될 때 정의되어야 하며 그 수가 너무 많아서도 안 된다. 테이블은 HBase에 의해 리전(Region)으로 자동 분할된다. 리전은 기본적으로 함께 저장된 인접한 범위의 로우들이다. 리전의 크기가 증가하여 임계값(설정 가능)을 넘어서게 되면 시스템에 의해 동적으로 분할된다. 최초에는 테이블에 리전이 단 하나 존재하지만, 데이터를 추가함에 따라 리전이 증가한다. 각 리전은 딱 하나의 리전 서버에서만 운용된다. <그림 7>은 로우들이 리전으로 그룹핑 되고 각기 다른 리전 서버에서 운용되는 것을 보여준다.



<그림 7> HBase 테이블 – 논리적 뷰

HBase에는 마스터 서버와 다수의 리전 서버가 주요 구성 요소로 존재하는데, 마스터 서버는 리전 서버 간에 리전의 부하 분산을 처리하는 역할을 담당한다. 과부하가 걸린 서버의 부하를 덜어주고 여유 공간이 많은 서버에 리전을 옮기기도 하며, 클러스터의 상태를 유지한다. 따라서 실제로 마스터 서버에 걸리는 부하는 가볍다. 리전 서버는 자신이 담당하는 모든 리전에 대한 읽기와 쓰기 요청을 처리하고, 임계값을 초과한 리전을 분할하기도 한다. <그림 8>은 HBase 구성 요소들의 조직도이다.



<그림 8> HBase 구성 요소 조직도

Ⅳ. 구현 및 테스트

HBase API는 자바 기반 API를 지원하기 때문에 JSP 내에 HBase API를 시용하기 용이하며 클라이언트의 요청에 따라 동적인 결과를 반환해야하므로 본 논문에서는 JSP를 이용하여 웹모니터링 페이지를 구축하였다. <그림 9>는 본 논문에 제안한 HBase 테이블을 설계한 논리적인 구조이다. TS는 Timestamp의 약자이며, 각 컬럼에는 분진, 이산화탄소, 조도, 온도, 습도 순서대로 나타내었다. 로우 키는 국가, 지역, 장소, 시간을 모두 포함시켜 최대 국가 단위까지의 넓은 범위의 데이터를 저장하도록 설계하였다. 로우 키에 많은 정보를 저장함으로써 가늘고 긴 테이블 모델을 따른다.

Row-id	Column Family									
국가 - 지역 - 장소명 - 시간	Dust	TS	CO_2	TS	Light	TS	Temp	TS	Hum	TS

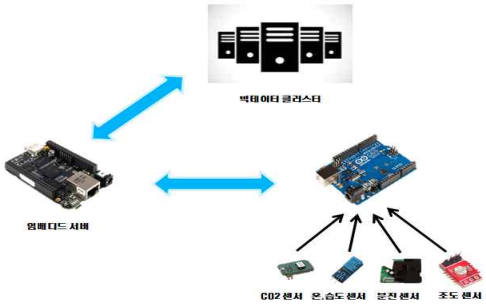
<그림 9> 테이블의 논리적 구조

그러나 본 논문에서는 사용가능한 하드웨어 자원의 한계와 장소의 제한으로 인해 <그림 10>과 같이 HBase 테이블을 설계하였다.

Row-id	Column Family									
시간	Dust	TS	CO_2	TS	Light	TS	Temp	TS	Hum	TS

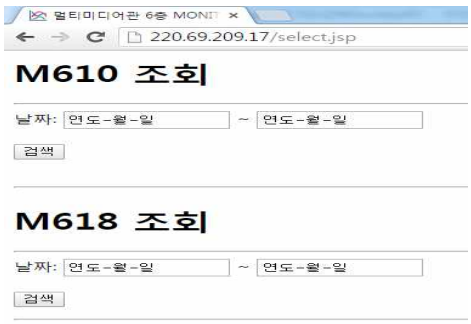
<그림 10> 실제 테이블의 논리적 구조

빅데이터 클러스터는 클라이언트 프로그램을 실행하여 실내에 설치된 임베디드 서버와 소켓 통신으로 센서 데이터를 수신한다. 클라이언트 프로그램은 백그라운드에서 실행되며, HBase에서 제공하는 Java API를 이용해 HBase 테이블에 센서 데이터를 저장한다. 본 논문에서 구현한 부분은 빅데이터 클러스터이며, 센서 데이터가 수집되는 과정은 <그림 11>에서 보여주고 있다.



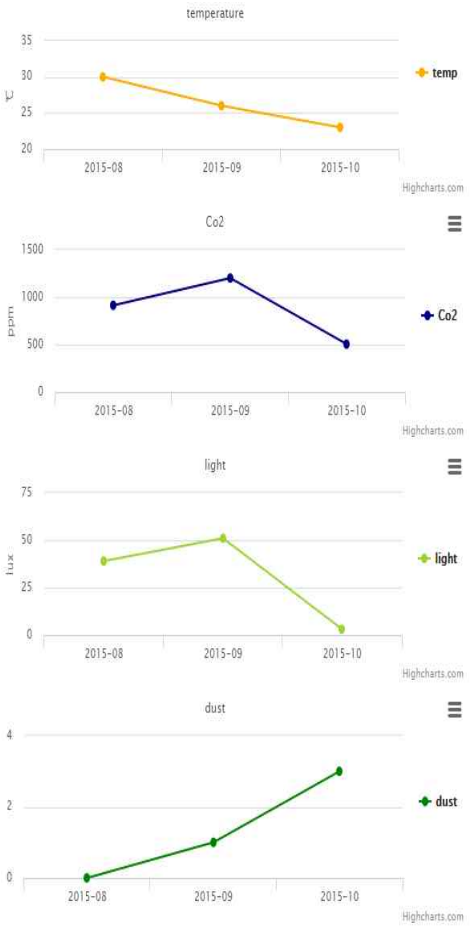
<그림 11> 센서 데이터 수집 과정

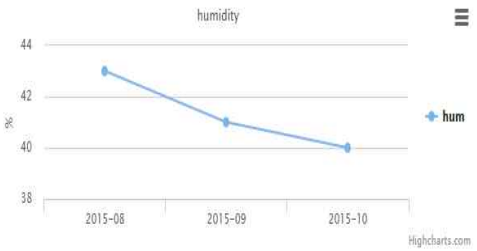
<그림 11>과 같이 수집된 데이터는 HBase에서 제공하는 Java API 중 하나인 Put 메소드를 이용하여 저장한다. Put 인스턴스를 생성하기 위해서는 로우 키를 제공해야 하며, 인스턴스 생성 후에는 add 메소드를 이용하여 컬럼을 하나씩 추가할 수 있다. 만약 타임스탬프 인수와 함께 사용하면 셀 단위로 데이터를 추가 할 수 있으며, 타임스탬프를 명시하지 않으면 해당 컬럼의 타임스탬프를 리전 서버에 위임하게 된다. 센서 데이터는 5초 단위로 저장되기 때문에 모든 센서 데이터를 모니터링 하기에는 무리가 있다. 따라서 본 논문에서는 빅데이터 클러스터에 저장된 센서 데이터를 클라이언트가 지정한 날짜 범위 내에서 월 단위로 센서 데이터의 값을 평균으로 계산하여 차트로 나타내었다. <그림 12>는 데이터 조회 페이지 요청 과정을 보여주고 있다.



<그림 12> 데이터 조회 페이지

<그림 13>에서는 실내 강의실인 날짜 범위(3개월)를 지정하여 검색을 한 온도, 습도, 조도, 먼지, 이산화탄소 수치를 월 평균을 차트로 나타내고 있다. 각 센서들은 모두 강의실 설치되어 있으며 온도의 경우는 계절이 겨울로 바뀌어가면서 점점 감소되고 있는 추세를 보여주고 있다. 또한, CO2 농도의 경우 학생들이 많이 움직이는 개강시기인 9월에 가장 높게 나타내고 있다. 먼지의 경우는 9월 개강 후 수업이 진행됨에 따라 분필가루와 각종 먼지들이 쌓이면서 수치가 높아지고 있다. 습도의 경우도 온도와 마찬가지로 점차 낮아지는 과정을 보여주고 있다.





<그림 13> 월 평균 차트

Ⅴ. 결론

본 논문에서는 빅데이터 클러스터를 구축하여 실내 환경 데이터를 수집하고, 모니터링을 통해서 실내의 쾌적한 환경을 유지할 수 있는 시스템을 설계 구현하였다. 본 논문의 실내 환경 모니터링 시스템은 2개의 강의실을 가지고 구현하였으나 모니터링이 가능한 공간이 확장 될 경우 지역, 국가 단위로 확장 될 수 있다. 따라서 수집되는 데이터의 단위도 거대해지므로 빅데이터 처리가 가능한 하둡 분산 파일 시스템과 HBase가 설치된 빅데이터 클러스터를 활용하여 데이터를 처리하고 모니터링이 가능한 시스템을 구현하였다. 현재는 데이터를 수집하고 모니터링이 가능하지만, 향후에는 데이터를 분석하여 분석 결과를 반환하고, 결과에 따른 이벤트 동작을 수행하도록 할 예정이다. 본 논문은 실시간 모니터링과는 다르게 누적된 히스토리 데이터를 모니터링하기 때문에 데이터 분석이 가능하고, 분석된 데이터 결과를 통해 인구 밀집도, 이동 동선 및 패턴 분석이 가능하며, 해당 공간에 대한 실내 환경 유지 및 개선이 가능하다. 또한 임계값을 정하여 반복된 패턴의 학습을 통해 이벤트 동작을 수행하도록 할 수 있다. 앞으로 최종적으로 분석까지 가능하게 된다면 조금 더 완성도 높은 시스템을 구현 할 수 있을 것으로 보인다.

참고문헌

[1] LG CNS, “미래 경영을 위한 빅데이터와 고급분석,” 2012.

[2] 김보람, 박수현, 홍봉희, “실내 환경 모니터링 시스템,” 한국정보과학회 2012 한국컴퓨터 종합학술대회 논문집, 제 39권, 제 1호(B), 2012, pp. 57-59.

[3] 장예슬, 조중현, 김기영, 이상정, “REST 서비스 기반 강의실 환경 모니터링 시스템,” 한국지식정보기술학회 추계학술대회논문집, 2015년 10월.

[4] 이응용, “빅데이터,” Net Term, 2013.

[5] 이현종, “빅데이터 하둡 플랫폼의 활용,” 한국통신학회지(정보와통신), 제 29 권, 제 11호, 2012, pp. 43-47.

[6] 조성우, “Big Data 시대의 기술,” KT 종합기술원, 2011.

[7] 톰 화이트, 하둡 완벽 가이드, 한빛미디어, 2013.

[8] 이현남, “하둡으로 배우는 대용량 데이터 분산처리 기술,” 지앤선 테크니컬 세미나, 2013.

[9] 서상원, 김재홍, 박윤성, 이준섭, 명재석, 대용량 데이터 분석 및 처리를 위한 Hadoop& NoSQL, 길벗, 2013.

[10] Ian Varley, “HBase Schema Design,” HBaseCon, 2012.

[11] 김정준, 한상근, 신인수, 한기준 “공간 HBase: 공간 빅 데이터를 위한 HBase 확장,” 정보과학회논문지: 데이터베이스, 제 40권, 제 5호, 2013, pp. 295-304.

[12] 라스 조지, HBase 완벽 가이드, 한빛미디어, 2013.

■ 저자소개 ■

전 병 찬
(Jeon Byoungchan)

2002년 3월~현재
청운대학교 기초교양교육원 교수
2001년 2월 순천향대학교 전산학과(공학박사)

관심분야 : 컴퓨터구조, 홈 네트워크, 모바일, 마이크로 프로세서 등
E-mail : jbc66@chungwoon.ac.kr

고 민 구
(Go Mingu)

2016년 2월 순천향대학교 전산학과 졸업

관심분야 : 홈 네트워크, 모바일, 무선 센서 네트워크
E-mail : mgk@naver.com

논문접수일 : 2017년 05월 19일
수 정 일 : 2017년 06월 05일
게재확정일 : 2017년 06월 05일