KIPS Trans. Comp. and Comm. Sys.
Vol.10, No.10 pp.277~284
ISSN: 2287-5891 (Print), ISSN 2734-049X (Online)

https://doi.org/10.3745/KTCCS.2021.10.10.277

Apriori Based Big Data Processing System for Improve Sensor Data Throughput in IoT Environments

Song Jin Su† · Kim Soo Jin† · Young Tae Shin††

ABSTRACT

Recently, the smart home environment is expected to be a platform that collects, integrates, and utilizes various data through convergence with wireless information and communication technology. In fact, the number of smart devices with various sensors is increasing inside smart homes. The amount of data that needs to be processed by the increased number of smart devices is also increasing, and big data processing systems are actively being introduced to handle it effectively. However, traditional big data processing systems have all requests directed to cluster drivers before they are allocated to distributed nodes, leading to reduced cluster-wide performance sharing as cluster drivers managing segmentation tasks become bottlenecks. In particular, there is a greater delay rate on smart home devices that constantly request small data processing. Thus, in this paper, we design a Apriori-based big data system for effective data processing in smart home environments where frequent requests occur at the same time. According to the performance evaluation results of the proposed system, the data processing time was reduced by up to 38.6% from at least 19.2% compared to the existing system. The reason for this result is related to the type of data being measured. Because the amount of data collected in a smart home environment is large, the use of cache servers plays a major role in data processing, and association analysis with Apriori algorithms stores highly relevant sensor data in the cache.

Keywords : IoT, Smart home, Apache Spark, Redis, Association Algorithm

IoT 환경에서 센서 데이터 처리율 향상을 위한 Apriori 기반 빅데이터 처리 시스템

송 진 수† · 김 수 진† · 신 용 태††

요 약

최근 스마트 홈 환경은 무선 정보통신 기술과 융합을 통해서 다양한 데이터를 수집·통합·활용하는 플랫폼이 될 것으로 전망되고 있으며 실제로 스마트 홈 내부에는 다양한 센서를 탑재한 스마트 디바이스 수가 점점 증가하고 있다. 증가된 스마트 디바이스 수만큼 처리해야하는 데이터의 양도 증가하고 있으며 이를 효과적으로 처리하기 위해 빅데이터 처리 시스템이 활발하게 도입되고 있다. 그러나 기존 빅데이터 처리 시스템은 분산 노드에 할당되기 전 모든 요청이 클러스터 드라이버로 향하기 때문에 동시에 많은 요청이 발생하는 경우 분할 작업을 관리하는 클러스터 드라이버에 병목현상이 발생하고, 이는 네트워크를 공유하는 클러스터 전체의 성능감소로 이어진다. 특히 작은 데이터 처리를 지속해서 요청하는 스마트 홈 디바이스에서 지연율이 더 크게 나타난다. 이에 본 논문에서는 동시에 다수의 센서에서 요청이 발생하는 스마트 홈 환경에서 효과적인 데이터 처리를 위한 Apriori 기반 빅데이터 시스템을 설계하였다. 제안하는 시스템의 성능평가 결과에 따르면, 데이터 처리 시간은 기존 시스템에 비해 최소 19.2%에서 최대 38.6% 단축됐다. 이러한 결과가 발생한 이유는 측정되는 데이터의 형태와 관련이 있다. 스마트 홈 환경은 수집되는 데이터의 양은 방대하나 각 데이터의 용량은 작기 때문에 캐시 서버의 사용이 데이터 처리에 큰 역할을 하며, Apriori 알고리즘을 통한 연관도 분석으로 사용자의 행동 습관과 연관도가 높은 센서 데이터를 캐시에 저장하기 때문에 캐시 서버의 활용률이 매우 높다.

키워드 : 사물인터넷, 스마트 홈, 아파치 스파크, 레디스, 연관도 분석 알고리즘

1. 서 론

4차 산업혁명의 기술이 발전하며 인공지능, 사물인터넷, 클라우드 컴퓨팅, 빅데이터와 같은 정보통신기술과 공간의 융합에 대한 연구가 활발하게 진행되고 있다. 사물 인터넷과 인공지능이 접목된 주거시설인 스마트 홈 시스템은 집 내부의 모든 디바이스가 네트워크로 연결되어 거주자의 생활패턴에 맞춰 실내안전, 전력관리, 헬스케어 등 다양한 분야에서 도움을 준다[1]. 초연결화를 근간으로 한 스마트 홈 구성을

※ 이 논문은 2017년도 정부(과학기술정보통신부)의 재원으로 정보통신기술진흥센터의 지원을 받아 수행된 연구임(No. 2017-0-00724, 셀룰러 기반 산업 자동화 시스템 구축을 위한 5G 성능 한계 극복 저지연, 고신뢰, 초연결 통합 핵심기술 개발).
※ 이 논문은 2021년 한국정보처리학회 춘계학술발표대회의 우수논문으로 "스마트홈 환경에서 센서 데이터 처리율 향상을 위한 기계학습 기반 캐싱 시스템 설계"의 제목으로 발표된 논문을 확장한 것임.
† 준 회 원 : 숭실대학교 컴퓨터학과 석사과정
†† 종신회원 : 숭실대학교 컴퓨터학부 교수
Manuscript Received : June 23, 2021
First Revision : July 29, 2021
Accepted : August 9, 2021
* Corresponding Author : Young Tae Shin(shin@ssu.ac.kr)

※ This is an Open Access article distributed under the terms of the Creative Commons Attribution Non-Commercial License (http://creativecommons.org/ licenses/by-nc/3.0/) which permits unrestricted non-commercial use, distribution, and reproduction in any medium, provided the original work is properly cited.

위해 스마트 홈 내부에 센서를 탑재한 디바이스가 증가하고 있으며, 각각의 디바이스는 실시간으로 방대한 양의 데이터를 생성한다[1,2]. 수집된 데이터는 빅데이터 처리 시스템을 통해 데이터의 연관관계를 도출하여 지능화된 스마트 홈 구축에 활용되고 있다.

빅데이터 처리 시스템은 분산 클러스터 구조로 연결되어 있으며 실질적인 데이터 처리는 클러스터 드라이버를 통해 분산된 노드에서 이루어진다. 그러나 분산노드에 할당되기 전 모든 요청이 클러스터 드라이버로 향하기 때문에 동시에 많은 요청이 발생하는 경우 분할 작업을 관리하는 클러스터 드라이버에 병목현상이 발생하고, 이는 네트워크를 공유하는 클러스터 전체의 성능감소로 이어진다. 특히 작은 데이터 처리를 지속적으로 요청하는 스마트 홈 디바이스에서 지연율이 더 크게 발생한다. 따라서 동시에 다수의 센서에서 요청이 발생하는 스마트 홈 환경에서 효과적인 데이터 처리 시스템이 필요하다.

이에 본 논문에서는 스마트 홈 환경에서 센서 데이터 처리율 향상을 위한 Apriori 알고리즘 기반 빅데이터 시스템을 제안한다. 제안하는 시스템은 기계학습 알고리즘을 통해 거주자의 생활패턴을 파악하고 사용되는 센서 간의 유사도를 측정하여 유사도가 높은 센서 그룹의 데이터를 캐시 서버에 저장한다. 연관도가 높은 데이터를 보유한 캐시 서버는 센서와 근접한 위치에 존재하여 데이터 반환 시간이 단축되고 반복된 빅데이터 서버의 호출을 줄여 시스템 전체의 효율성을 증대시킨다.

본 논문의 구성은 다음과 같다. 2장에서는 기존에 분산 클러스터 기반으로 사용 중인 인-메모리 빅데이터 처리 시스템과 비관계형 데이터베이스 중 인-메모리로 구성된 데이터베이스를 살펴보고 시계열 데이터의 연관도 분석 기법에 대해 살펴본다. 3장에서는 본 논문에서 제안하는 스마트 홈 환경에서 센서 데이터 처리율 향상을 위한 기계학습 기반 캐싱 시스템을 제시한다. 4장에서는 제안한 시스템의 성능을 분석하고, 마지막 5장에서는 결론 및 향후 연구 과제를 제시한다.

2. 배경 및 관련 연구

본 장에서는 IoT 환경에서 가장 보편적으로 사용되는 빅데이터 처리 시스템과 데이터베이스를 연구한다. IoT환경에서 구축되는 시스템은 데이터의 실시간 처리가 중요하기 때문에 인-메모리 기반으로 동작하는 빅데이터 분산 처리 시스템과 데이터베이스를 살펴본다. 또한 연구된 결과를 기반으로 제안하는 기계학습 알고리즘 기반 캐싱 시스템의 요구사항을 도출한다.

2.1 인-메모리 기반 분산 시스템 아파치 스파크

스파크는 빅데이터 처리를 위한 인-메모리 기반의 병렬분산 처리 시스템이다. 스파크는 데이터를 메모리에서 처리하기 때문에 기존의 디스크 기반으로 동작하는 빅데이터 시스템에 비해 데이터 처리 속도가 최소 10배에서 100배이상 단축되어 실시간 빅데이터 처리에 보편적으로 사용되고 있다[2]. Fig. 1은 스파크의 구성이다.

스파크는 인프라, 스파크 코어, 라이브러리로 구성되어 있다. 인프라는 스파크 실행 및 스케줄러 리소스 관리를 하고 그 위로 메모리 기반의 분산 클러스터 컴퓨팅 환경의 스파크 코어가 실행된다. 스파크는 라이브러리를 통해 실시간 데이터 처리를 용이하게 하고, 머신러닝 알고리즘을 활용할 수 있다. 스파크가 기본 제공하는 라이브러리는 빅데이터를 SQL로 처리하는 Spark SQL, 실시간으로 전송되는 스트리밍 데이터를 처리하는 Spark Streaming, 머신러닝을 위한 MLib, 그래프 데이터 프로세싱이 가능한 GraphX이 있다[3]. 스파크 클러스터의 구조는 드라이버, 클러스터 매니저, 워커 노드로 이루어져 있다. Fig. 2는 스파크의 클러스터 구조를 나타낸다.

스파크는 드라이버 프로세스와 다수의 워커 노드가 네트워크로 연결되어 있는 분산 처리 시스템이다. 드라이버 프로세스는 입력에 대한 응답, 워커 노드의 작업과 관련된 스케줄링 역할을 수행하고 워커 노드에서는 드라이버 프로세스가 할당한 작업을 수행한다.

클러스터 매니저는 스파크 컨텍스트와 워커 노드 사이에 중계자 역할을 하며 다수의 워커 노드가 작업을 공유할 수 있게 한다[3]. 워커노드는 요청된 작업을 수행하기 위해 다수의 태스크를 워커 노드에 분산하여 처리한다.

스파크의 데이터 처리는 클러스터의 구조와 밀접한 연관이 있어 클러스터 상에서 동작하는 스파크 컨텍스트와 워커 노드의 데이터 송수신으로 인한 오버헤드를 줄일수록 성능이 증가한다.

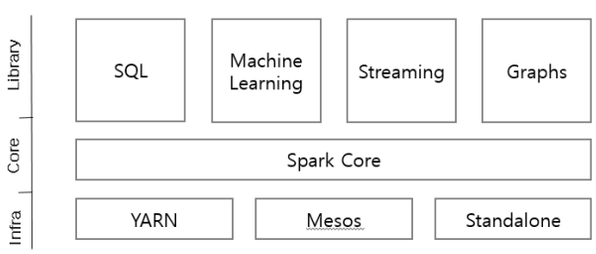

Fig. 1. A Spark Configuration

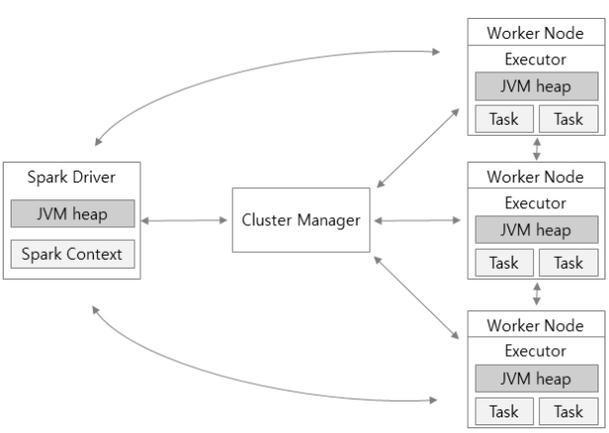

Fig. 2. A Spark Cluster Structure Diagram

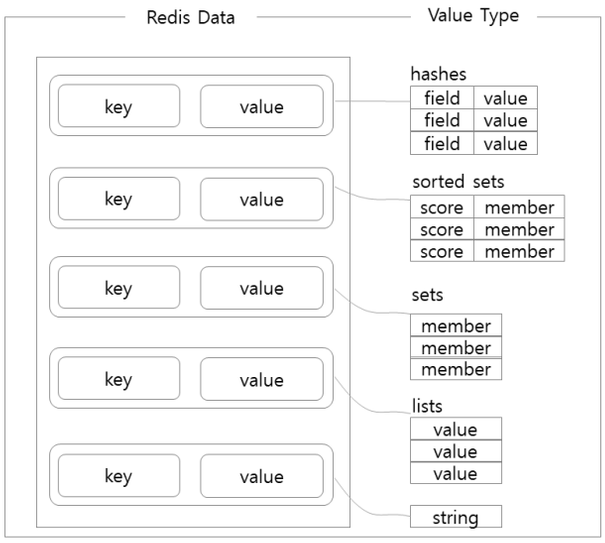

Fig. 3. Redis Data Structure

2.2 인-메모리 기반 데이터베이스 레디스

레디스는 키-값 구조로 데이터를 관리하는 비관계형 데이터베이스이다. 레디스는 모든 데이터를 메모리에 저장하는 데이터베이스로 일반 디스크 저장소를 사용하는 데이터베이스에 비해 빠른 속도로 데이터를 처리한다. 레디스의 성능은 서버의 품질에 따라 다르나 평균 초당 2만-10만회를 수행하며, 빠른 Read, Write 속도를 보장한다. 다음 Fig. 3은 레디스의 데이터 구조를 나타낸다.

레디스가 지원하는 데이터 구조는 hashes, sorted sets, sets, lists, string으로 크게 5가지이다. 레디스는 비관계형 데이터베이스이며, 다양한 형식의 데이터를 저장하고 처리할 수 있어 사용자들의 대규모 메시지를 실시간으로 처리 하는 인스타그램, LINE, StackOverflow, Vlizzard 등 많은 소셜 서비스에서 사용되고 있다[4,5].

2.3 연관도 분석 기법

머신러닝은 학습 시스템에 따라 지도학습, 비지도학습, 강화학습 세 가지로 분류한다. 그 중 비지도 학습 알고리즘인 연관성 규칙은 다수의 품목 간의 관계를 수치화하여 연관규칙을 찾아내는 기법이다. 기존의 데이터를 특별한 변형 없이 사용할 수 있어 다양한 분야에서 두 개 이상의 품목 간의 관련성과 규칙을 탐색하는데 활용된다. 연관규칙의 평가기준은 지지도, 신뢰도, 향상도가 있다. 지지도는 항목집합 X와 Y가 동시에 발생할 비율을 의미하며, Equation (1)과 같이 정의된다.

$$Support_{(X \Rightarrow Y)} = (X \cap Y) \tag{1}$$

신뢰도는 항목집합 X가 포함된 비율 중 항목집합 X와 Y가 동시에 포함된 비율을 의미하며, Equation (2)와 같이 정의된다.

$$Confidence_{(X \Rightarrow Y)} = P(Y \mid X) = \frac{P(X \cap Y)}{P(X)} \tag{2}$$

향상도는 실제 발생 확률을 각 항목집합의 발생이 독립적일 경우 그 거래가 동시에 발생할 예상 기대확률로 나눈 것을 의미하며, Equation (3)과 같이 정의된다.

$$Lift_{(X \Rightarrow Y)} = \frac{P(Y \mid X)}{P(Y)} = \frac{P(X \cap Y)}{P(X)P(Y)} \tag{3}$$

향상도의 값이 크면 두 항목이 동시에 발생한 확률이 예상 확률보다 더 크므로 향상도의 값이 높은 경우에 의미 있는 규칙이다. 최소 지지도 값과 최소 신뢰도 값을 모두 만족하고 향상도의 값이 높은 경우 두 항목집합의 규칙을 강한 연관규칙으로 판단한다[6].

2.4 요구사항 도출

본 절에서는 앞서 설명한 빅데이터 처리 시스템의 처리 방식에 대한 문제점을 도출하고, 요구사항을 분석한다.

빅데이터 처리 시스템인 스파크는 분산 클러스터 환경으로 구성되어 있다. 스파크는 데이터 처리 요청에 대해 분산된 워커 노드에서 실제 작업을 수행하며, 워커노드에 작업을 분배할 때 드라이버에 의해 태스크 단위로 할당된다. 그러나 작은 데이터 처리를 지속적으로 요청하는 스마트 홈 환경에서는 태스크가 너무 많이 생성되어 드라이버의 태스크 할당 작업에 병목현상이 발생하고, 이는 네트워크 자원을 공유하는 클러스터 전체의 성능 감소로 이어진다. 따라서 빅데이터 서버에서 반복된 데이터를 요청하는 횟수를 줄여 드라이버의 태스크 할당 단계에서 발생하는 병목현상을 해소하고 빅데이터 서버의 자원을 효과적으로 사용하는 것이 중요하다.

3. 제안하는 빅데이터 처리 시스템

제안하는 빅데이터 처리 시스템은 기존 빅데이터 분석 시스템의 요구사항에 맞춰 설계하며 스파크의 단점을 보완하기 위해 빈번하게 사용되는 데이터와 연관도가 높은 데이터를 예측하여 센서와 가까운 캐시서버에 미리 저장한다. 캐시 서버에 사용되는 연관도 분석 알고리즘은 거주자의 위치, 사용된 센서, 시간을 분석하기 때문에 캐시 메모리의 세 가지 지역성인 시간 지역성, 공간 지역성, 순차 지역성을 모두 만족하며 캐시 서버의 활용률을 향상시킨다. 캐시 서버의 활용률이 향상되면 반복된 빅데이터 서버의 호출이 줄어 시스템 전체의 효율성이 증대된다.

3.1 제안하는 시스템 구성

제안하는 빅데이터 처리 시스템은 인-메모리 기반 분산 클러스터 아파치 스파크를 활용하여 스마트 홈 디바이스에서 생성된 데이터를 수집하고, 수집된 시계열 데이터에 연관도 분석 알고리즘을 적용하여 시간 별로 빈번하게 요청되는 센

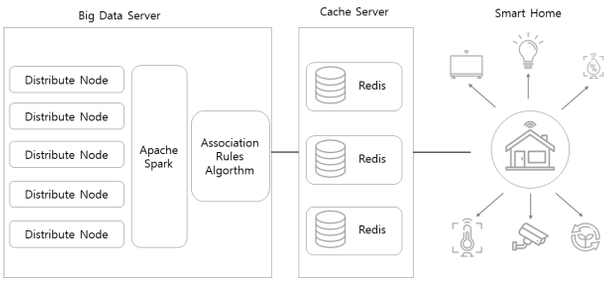

Fig. 4. A Schematic Diagram of the Proposed Technique

서와 연관성이 높은 센서의 데이터를 인-메모리 기반 데이터베이스 레디스에 저장한다. 제안하는 시스템은 스마트 홈 디바이스를 기준으로 빅데이터 서버와 캐싱 서버로 구성된다. Fig. 4는 제안하는 빅데이터 처리 시스템의 구조도를 나타낸다.

빅데이터 서버는 거주자 생활패턴 분석에 필요한 데이터를 수집하는 영역으로 스마트 홈 센서에서 발생한 데이터를 수집 및 저장하는 기능을 수행한다. 스파크 드라이버와 분산 노드로 이루어진 클러스터 환경으로 구성하며 수집된 데이터는 분산노드에 저장한다.

연관도 분석 알고리즘은 빅데이터 서버와 연동하여 스마트 홈 센서에서 수집된 데이터의 연관도 분석을 통해 항목 별 연관도가 높은 순으로 캐시 서버에 저장하는 기능을 수행한다.

캐시 서버는 연관도 분석 결과 높은 연관도를 가진 센서 그룹의 데이터를 미리 저장하고, 빅데이터 서버와 센서의 사이에 위치하여 시간 별로 센서의 요청에 대한 빠른 처리를 담당하는 역할을 수행한다. 스마트 홈 센서에서 보내온 요청 중에 예측된 데이터가 존재하면 캐시 서버에서 바로 처리한다. 빅데이터 서버는 클러스터 드라이버가 FIFO 방식으로 스케줄링 되기 때문에 실시간으로 발생하는 센서의 데이터를 지속적으로 수집하고 있다. 따라서 데이터 탐색을 요청하는 작업은 데이터 수집 작업을 수행중인 빅데이터 서버에 할당하는 것 보다 캐시 서버에서 처리하고 반환하는 것이 빠르다.

3.2 연관도 분석 알고리즘

연관도 분석 알고리즘은 어떤 센서 집합이 빈번히 발생하는가를 알려주는 일련의 규칙을 생성한다. 제안하는 시스템에서는 거주자의 IoT센서 디바이스의 사용 이력을 기반으로 “X 디바이스를 사용한 고객들은 Y 디바이스를 사용할 가능성이 크다”는 사용 순서에 존재하는 규칙을 찾아내는 역할을 한다. 연관성 규칙은 센서 4개만 가지고 규칙을 생성하더라도 조합된 규칙의 수가 약 50개가 되기 때문에 무의미한 데이터 셋을 제거해야 규칙 생성 시간을 단축할 수 있다. 따라서 본 논문의 알고리즘 실행 단계는 데이터 전처리 단계와 연관규칙 생성 단계로 이루어져 있다.

1) 전처리 단계

IoT 디바이스에서 수집된 데이터 셋은 센서가 동작 중일 때 지속해서 로그데이터를 생성하여 단시간에 중복된 데이터가 대량으로 발생한다. 연관도 분석 알고리즘은 빈발항목 집합에서 “X센서를 사용한 후 Y센서를 사용했다”는 규칙을 찾기 때문에 센서가 동작하는 시간 동안 생성된 모든 데이터가 아닌 어떤 센서 항목이 동작했는지만 활용된다. 데이터 전처리 단계를 통해 Table 1의 원시 데이터에서 센서 이름과 측정된 시간을 추출하고 일정 시간 동안 사용된 센서의 중복된 데이터를 제거하여 함께 사용된 센서를 그룹으로 병합한다. Table 2와 같이 빈발항목 집합 생성을 위한 데이터 셋으로 정제한다.

Table 1. Example of a Raw Dataset

value_id	sensor_id	timestamp	value
18730541	5892	00:00.1	0
18730542	5887	00:00.1	1024
18730544	5891	00:00.2	0
18730546	5896	00:00.4	652
18730548	6127	00:00.5	1024
18730549	5888	00:00.5	0
18730550	5889	00:00.6	301
18730551	5893	00:00.6	0
18730553	5895	00:00.7	0
18730554	5894	00:00.7	652

Table 2. Example of a Transform Dataset

timestamp	sensor_id
2020-02-26 11	{ 6222, 6223 }
2020-02-26 19	{ 6223, 6222 }
2020-03-12 14	{ 6425, 3214, 6124 }
2020-03-12 17	{ 6481, 5832 }
2020-05-03 13	{ 5892, 5894 }
2020-06-17 12	{ 5992, 6222 }
2020-07-15 14	{ 6344, 5851, 4374 }
2020-08-26 10	{ 6127, 6220, 6686 }
2020-08-26 13	{ 5892, 5894 }

2) 연관규칙생성

연관규칙은 전처리 된 데이터 셋 각각의 센서 지지도를 계산하고 알고리즘에 정의된 최소지지도보다 크거나 같은 조건을 만족하는 데이터로 빈발항목 집합을 구성한다. 발견된 규칙은 지지도, 신뢰도, 향상도를 분석하여 규칙의 정확성을 고려한다. 위 3개의 값을 계산하기 위한 알고리즘은 다음 Fig. 5와 같고 알고리즘을 적용한 결과는 Table 3과 같다.

Fig. 5의 알고리즘을 적용하여 규칙을 생성한다. 약 7만9천 건의 센서 정보에서 최소 지지도를 넘는 규칙은 약 296건이며, 규칙을 생성 하는데 사용된 시간은 123ms이다. 규칙을 적용한 결과는 Table 4와 같이 연관도가 높은 센서 디바이스의 순서대로 정렬하고 정렬된 추천데이터는 함께 사용되는 빈도가 높은 거로 판단되어 SensorX 데이터를 캐시 서버에 적재 시 Sensor Y 데이터를 함께 적재한다.

Association Analysis Algorithm

```
# minimum support, associated rule return function
def association_rules(device_sensor, min_support):
# support, frequency calculation
sensor_stats=freq(device_sensor).to_frame("freq")sen
sor_stats['support'] = sensor_stats['freq'] /
device_count(device_sensor) * 100
# exclude less than mininum support
qualifying_sensors =
sensor_stats[sensor_stats['support'] >=
min_support].index order_sensor
=device_sensor[device_sensor.isin(qualifying_sensor
s)]
# exclude less than 2 information
device_size= freq(device_sensor.index)
qualifying_device= device_size[device_size >=
2].index device_sensor=
device_sensor[device_sensor.index.isin(qualifying_de
vice)]
# frequency, support calculation
sensor_stats =
freq(device_sensor).to_frame("freq")sensor_stats['sup
port'] = sensor_stats['freq'] /
device_count(device_sensor) * 100
# create generator
sensor_pair_gen=get_sensor_pairs(device_sensor)
# frequency of sensor sets, support calculation
sensor_pairs=freq(sensor_pair_gen).to_frame("freqA
B") sensor_pairs['supportAB'] =
sensor_pairs['freqAB'] / len(qualifying_device) * 100
#  if muninum support is exclude
sensor_pairs=
sensor_pairs[sensor_pairs['supportAB'] >=
min_support]
# generate metrics for calculated
sensor_pairs =
sensor_pairs.reset_index().rename(columns={'level_0
': 'sensor_A', 'level_1': 'sensor_B'}) sensor_pairs =
merge_sensor_stats(sensor_pairs, sensor_stats)
sensor_pairs['confidenceAtoB'] =
sensor_pairs['supportAB'] / sensor_pairs['supportA']
sensor_pairs['confidenceBtoA'] =
sensor_pairs['supportAB'] / sensor_pairs['supportB']
sensor_pairs['lift']                =
sensor_pairs['supportAB'] / (sensor_pairs['supportA']
* sensor_pairs['supportB'])
# sorting enhancements to return results
return sensor_pairs.sort_values('lift',
ascending=False)
```

Fig. 5. Apriori-based Association Algorithm

Table 3. Generated Association Rules

Starting sensor_data	79904
Sensors with support >=0.01	24
Remaining sensor_data	79904
Remaining sensors with 2+ items	4206
Remaining sensor_data	79904
sensor pairs	296
sensor pairs with support >= 0.01	296
time	123

Table 4. Recommended Sensor Data for the Proposed System

Sensor X	Sensor Y
bathroom/ambience/temperature	bathroom/ambience/humidity
bathroom/washingmachine/current	kitchen/coffeemaker/current
bathroom/washingmachine/current	kitchen/dishwasher/current
kitchen/sandwichmaker/current	kitchen/coffeemaker/current
bathroom/washingmachine/current	bathroom/ambience/temperature
kitchen/dishwasher/current	kitchen/coffeemaker/current
kitchen/sandwichmaker/current	kitchen/dishwasher/current
bathroom/washingmachine/current	kitchen/sandwichmaker/current
kitchen/kettle/current	kitchen/dishwasher/current
entrance/door/contact	bedroom/weightscale/pressure
kitchen/sandwichmaker/current	kitchen/kettle/current
kitchen/dishwasher/current	kitchen/kettle/current

4. 성능 평가

본 장에서는 스마트 홈에서 센서 데이터에 관한 요청이 발생할 때 제안하는 시스템이 기존 빅데이터 처리 방식보다 효과적임을 증명한다.

성능평가는 기존 시스템과 제안하는 시스템의 성능비교를 위해 성능평가 시나리오를 작성하고 거주자 행동 패턴 별로 처리 시간을 측정한다.

4.1 성능평가 시나리오

성능평가 시나리오는 센서 개수를 증가하며 동시에 요청되는 센서 데이터에 대한 처리 시간을 분석하는 것으로 데이터 요청 후부터 실제 데이터를 반환 받기까지의 시간을 측정한다. 사용되는 데이터의 양은 시나리오 Case별로 200MB씩 증가시킨다. 성능평가 시나리오는 Table 5와 같다.

성능평가 시나리오에 활용할 데이터 셋은 Orebro Universitet에서 제공한 Multi-sensor dataset of human activites in a smart home enviromnent이다. Fig. 6은 데이터 셋의 측정환경 및 센서 위치를 나타낸다.

데이터 셋의 측정환경은 아파트이며, 내부 구조는 발코니,

Table 5. The Performance Evaluation Scenario

	Location	Sensor Type, Count	Object	Total
Case1	Bathroom	Motion, 1	Ambience	1
Case2	Bathroom Living room	Motion, 1 Light, 1	Ambience, TV	2
Case3	Bathroom Living room Kitchen	Motion, 1 Light, 1 Smart plug, 1	Ambience, TV, Coffee maker	3
Case4	Bathroom Living room Kitchen Balcon	Motion, 1 Light, 1 Smart plug, 1 Reed switch. 1	Ambience, TV, Coffee maker, Door contact	4
Case5	Bathroom Living room Kitchen Balcon	Motion, 1 Light, 1 Smart plug, 1 Reed switch. 1 Temperature & Humidity, 1	Ambience, TV, Coffee maker, Door contact, Ambience	5
Case6	Bathroom Living room Kitchen Balcon Entrance Bedroom	Motion, 1 Light, 1 Smart plug, 1 Reed switch. 1 Temperature & Humidity, 1 Pressure, 1	Ambience, TV, Coffee maker, Door contact, Ambience, Bed	6
Case7	Bathroom Living room Kitchen Balcon Entrance Bedroom	Motion, 2 Light, 1 Smart plug, 1 Reed switch. 1 Temperature & Humidity, 1 Pressure, 1	Ambience, TV, Coffee maker, Dishwasher, Door contact, Ambience, Bed	7
Case8	Bathroom Living room Kitchen Balcon Entrance Bedroom	Motion, 2 Light, 2 Smart plug, 1 Reed switch. 1 Temperature & Humidity, 1 Pressure, 1	Ambience, TV, Stove, Coffee maker, Dishwasher, Door contact, Ambience, Bed	8
Case9	Bathroom Living room Kitchen Balcon Entrance Bedroom	Motion, 2 Light, 2 Smart plug, 2 Reed switch. 1 Temperature & Humidity, 1 Pressure, 1	Ambience, TV, Stove, Coffee maker, Dishwasher, Door contact, Ambience, Bed, Couch	9
Case10	Bathroom Living room Kitchen Balcon Entrance Bedroom	Motion, 3 Light, 2 Smart plug, 2 Reed switch. 1 Temperature & Humidity, 1 Pressure, 1	Ambience, TV, Stove, Coffee maker, Dishwasher, Door contact, Ambience, Bed, Couch, Fridge	10

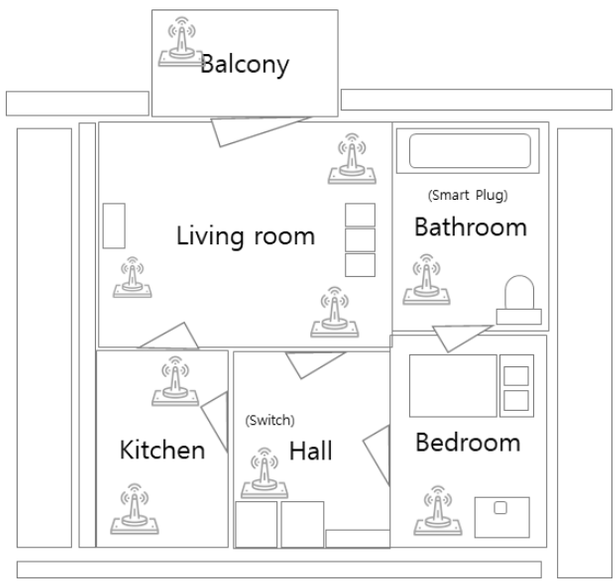

Fig. 6. A Schematic Diagram of the Proposed Technique

Table 6. The Environment for Performance Analysis

	CONTENT
CPU	Intell(R) Core(TM) i7-9700k 3.60Ghz 32G RAM SSD 500GB
RAM	32.0GB
HDD	SSD 500GB
OS	Linux CentOS 7
SW	Apache Spark 2.4.7 Redis 6.2.1
DATA	doi :10.17632/t9n68ykfk3.1 Institutions : Orebro Universitet Multi-sensor dataset of human activities in a smart home environment

욕실, 거실, 주방, 침실, 복도로 이루어져 있다. 스마트 홈에 배치된 센서는 수동 적외선 센서, 압력 감지 센서, 스위치, 광 센서, 온도 및 습도 센서, 스마트 플로그로 구성되어 거주자의 활동 데이터가 수집된다. 데이터는 침대, 소파, TV, 냉장고, 커피머신, 식기 세척기에 가해지는 압력과 소비 전류 등 사용자와 디바이스 간의 상호작용 데이터를 기록하고 있으며 2020년 2월 26일부터 2020년 8월 26일까지 6개월간 1Hz의 빈도로 총 12GB의 데이터가 수집되었다[7,8].

4.2 실험 환경

제안하는 빅데이터 처리 시스템의 성능을 분석하기 위한 실험환경은 스파크에 캐싱 서버를 추가하여 구성했다. 성능 평가는 기존 시스템과 제안하는 시스템의 데이터 처리 시간을 비교 분석한다. Table 6은 제안하는 빅데이터 처리 시스템의 성능평가를 위한 주요 환경 구성을 나타낸다.

Table 7. The Comparison of Present Scheme and Propose Scheme for Analysis Process Time

	Present(sec)	Propose(sec)	Reduction(%)
Case1	1.881782	1.244471	-33.87(%)
Case2	2.014556	1.298915	-35.52(%)
Case3	2.090438	1.319458	-36.88(%)
Case4	2.278805	1.398949	-38.61(%)
Case5	2.350087	1.472013	-37.36(%)
Case6	2.380895	1.530864	-35.70(%)
Case7	2.461262	1.675513	-31.92(%)
Case8	2.553200	1.814524	-28.93(%)
Case9	2.557146	1.913003	-25.19(%)
Case10	2.698255	2.178865	-19.25(%)

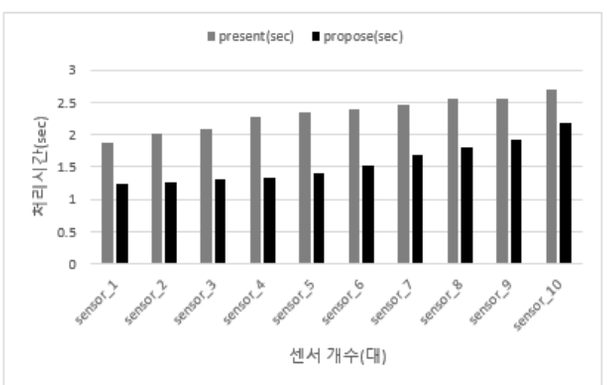

Fig. 7. The Comparsion of Present Scheme and Propose Sceme for Analysis Process Time

실험환경은 Intell(R) Core(TM) i7-9700k 3.60Ghz 32G RAM 500G SDD를 사용하며 Vmware를 활용해 가상환경을 구성한다. 가상환경에 구성한 스파크는 정식 배포된 2.4.7 버전을 사용하며 드라이버와 워커 노드가 단일 머신에서 실행되는 독립 모드로 구성한다.

4.3 기존 시스템과 제안하는 시스템 성능 비교

성능평가는 작동하는 센서의 개수를 증가시키며 처리 시간을 측정한다. 처리 시간은 거주자가 디바이스를 사용하기 위해 데이터를 요청한 후부터 서버 접속을 통해 데이터 반환이 완료되는 시간까지로 한다. Table 7과 Fig. 7은 테스트 시나리오에 따른 처리 시간을 비교한 결과를 나타낸다.

성능평가 결과에 따르면, 제안하는 시스템은 기존 시스템보다 데이터 처리 시간이 단축됨을 보인다. 데이터 처리속도는 제안하는 시스템이 기존 시스템보다 최소 19.2%에서 최대 38.6%로 빠르게 데이터를 처리하였으며, 기존의 시스템에서는 Case1 일 때에만 2초 이내에 처리시간이 걸렸던 반면에 제안한 시스템은 Case1-9까지 2초 이내에 처리했다.

5. 결 론

IoT 환경의 발전으로 네트워크로 연결된 다수의 센서에서 실시간으로 거주자의 활동정보를 수집하는 스마트 홈 환경은 효과적인 빅데이터 처리 시스템을 활용하기 위해 다수의 센서에서 생성되는 데이터를 효율적으로 처리하는 것이 중요하다.

이에 본 논문에서는 스마트 홈 환경에서 센서 데이터 처리율 향상을 위한 빅데이터 처리 시스템을 제안하였다. 제안하는 시스템은 스파크와 레디스를 결합하여 구성하고 연관도 분석 알고리즘을 통해 캐시 서버에 데이터를 적재한다. 제안하는 시스템의 성능평가 결과에 따르면, 데이터 처리 시간은 기존 시스템보다 최소 19.2%에서 최대 38.6% 단축됐다.

이러한 결과가 발생한 이유는 측정되는 데이터의 형태와 관련이 있다. 기존 빅데이터 서버는 처리하는 데이터 각각의 용량이 크기 때문에 캐시 서버를 구축하지 않는다. 하지만 스마트 홈 환경은 수집되는 데이터의 양은 방대하나 각 데이터의 용량은 작기 때문에 캐시 서버의 사용이 데이터 처리에 큰 역할을 한다. 또한 캐시 서버에 적재하는 데이터는 인공지능 알고리즘을 적용하여 사용자의 행동습관과 데이터의 연관성이 높기 때문에 데이터의 탐색 속도가 매우 높다.

향후 제안하는 시스템을 적용한 빅데이터 시스템의 메모리 사용량과 네트워크 사용량에 대한 보다 구체적인 성능분석이 필요하다. 또한, 실 환경을 대상으로 한 개발 및 구현을 통해 제안하는 기법의 보다 현실적인 검증이 필요하다.

References

[1] M. R. Alam, M. B. I. Reaz, and M. A. M. Ali, "A review of smart homes-past, present, and future," in *IEEE Transactions on Systems, Man, and Cybernetics, Part C (Applications and Reviews)*, Vol.42, No.6, pp.1190-1203, Nov. 2012, doi: 10.1109/TSMCC.2012.2189204.

[2] H. Lee, Y.-W. Kim, and K.-Y. Kim, "Study of in-memory based hybrid big data processing scheme for improve the big data processing rate," *Journal of Korea Institute of Information, Electronics, and Communication Technology*, Vol.12, No.2, pp.127-134, Apr. 2019.

[3] K. Ji and Y. Kwon, "Performance comparison of python and scala APIs in spark distribured cluster computing system," *Korea Multimedia Society*, Vol.23, No.2, pp.241-246, Feb. 2020.

[4] H. C. Park and K. H. Cho, "Waste database analysis joined with local information using association rules," *Journal of The Korean Data Analysis Society*, Vol.7, No.3, pp.763-772, 2005.

[5] J. M. Choi, D. W. Jeoung, J. S. Yoon, and S. J. Lee, "Digital forensics investigation of redisdatabase," *KIPS Transactions on Computer and Communication Systems*, Vol.5, No.5, pp.117-126, May 2016.

[6] B. M. Seo, B. S. Jang, H. S. Oh, and H. J. Park, "Restful, redis based API thin server platform design for automatic API generation and data processing performance," *The Journal of Korean Institute of Communications and Information Sciences*, Vol.44, No.5, pp.895-903. 2019.
[7] G. Chimamiwa, M. Alirezaie, F. Pecora, and A. Loutfi, "Multi-sensor dataset of human activities in a smart home environment," *Data in Brief*, Vol.34, pp.106632, 2021, https://doi.org/10.1016/j.dib
[8] G. Chimamiwa, M. Alirezaie, F. Pecora, and A. Loutfi, "Multi-sensor dataset of human activities in a smart home environment," Mendeley Data, V1, 2020, doi: 10.17632/t9n68ykfk3.1

송 진 수
https://orcid.org/0000-0002-5391-1866
e-mail : iko153@soongsil.ac.kr
2021년 ~ 현 재 숭실대학교 컴퓨터학과 석사과정
관심분야 : IoT, 데이터 분석, 인공지능, 클라우드 컴퓨팅

김 수 진
https://orcid.org/0000-0001-9223-8654
e-mail : soojk129@soongsil.ac.kr
2016년 동국대학교 컴퓨터학부(학사)
2021년 ~ 현 재 숭실대학교 컴퓨터학과 석사과정
관심분야 : 인공지능, 5G, 빅데이터, 클라우드 컴퓨팅

신 용 태
https://orcid.org/0000-0002-1199-1845
e-mail : shin@ssu.ac.kr
1985년 한양대학교 산업공학과(학사)
1990년 Univ. of lowa, 컴퓨터학과(석사)
1994년 Univ. of lowa, 컴퓨터학과(박사)
1995년 ~ 현 재 숭실대학교 컴퓨터학부 교수
관심분야 : 정보보호, 인터넷 프로토콜, IoT, 클라우드 컴퓨팅