한국컴퓨터정보학회 동계학술대회 논문집 제27권 제1호 (2019. 1)

IoT 노드 클러스터 기반의 실시간 스트림 데이터 처리 방안

임환희[O], 김동현[*], 이병준[*], 김경태[**], 윤희용[*]
[O*]성균관대학교 전자전기컴퓨터공학과
[**]성균관대학교 소프트웨어대학 소프트웨어학과
e-mail: {lhh423, kdh7263, byungjun}@skku.edu[O*], kyungtaekim76@gmail.com[**], youn7147@skku.edu[*]

Real-time stream data processing method based on IoT node cluster

Hwan-Hee Lim[O], Dong-Hyun Kim[*], Byung-Jun Lee[*], Kyung-Tae Kim[**], Hee-Yong Youn[*]
[O*]Dept. of Electrical and Computer Engineering, Sungkyunkwan University
[**]Dept. of Software, Sungkyunkwan University

● 요 약 ●

Edge Computing 환경에서는 데이터 처리와 시스템 제어를 위한 별도의 서버가 존재하지 않는다. 서버를 통한 중앙통제 방식이 아닌 Edge computing에 사용된 IoT기기들이 연동되어 데이터 분산 처리와 연산을 통해 전체 시스템이 동작된다. 이러한 Edge computing 시스템 구조 특성상 전체 시스템이 과부하를 피하기 위해 각 IoT 기기에서 동시다발적으로 감지되는 실시간 상황 정보를 효율적으로 처리 하여야한다. 이에 따라 실시간 상황 정보를 효율적으로 처리하거나, 다양한 데이터 분석처리 알고리즘들이 연구 개발되어 데이터 처리에 적용되어 왔다. 하지만 데이터의 정보 흐름과 타입에 초점을 맞춘 것이 아니라 예상분석 및 획일화된 알고리즘을 통해서 분석되기 때문에 해당 플랫폼이 주로 지향하는 데이터 형식에 맞지 않으면 성능저하를 수반하며 사용에 제약이 많은 문제점이 있다. 따라서 본 논문에서는 IoT 환경에서 실시간 반응성 향상을 목표로 오픈소스 기반 스트림 데이터 처리 방법에 대한 비교 분석과 Fast-reaction을 위한 데이터 처리 도구 비교 분석을 연구를 진행한다.

키워드: Edge Computing, IoT, 스트림 데이터, 데이터 분산 처리

I. Introduction

Edge computing 환경에서는 데이터 처리와 시스템 제어를 위한 별도의 서버가 존재 하지 않는다 [1]. 서버를 통한 중앙통제 방식이 아닌 Edge computing에 사용된 IoT기기들이 연동되어 데이터 분산 처리와 연산을 통해 전체 시스템이 동작된다. 이러한 Edge computing 시스템 구조 특성상 전체 시스템이 과부하를 피하기 위해 각 IoT 기기에서 동시다발적으로 감지되는 실시간 상황 정보를 효율적으로 처리 하여야한다. 그리고 Edge computing에서 군집 지능화를 위해서 다양한 데이터 패턴에 적합한 기계 학습 기법이 필수적이나 기존 하둡(Hadoop)[3]의 맵리듀스(Map reduce)[4]는 배치 방식으로 데이터를 처리하기 때문에 실시간으로 데이터를 처리 및 분석하기 어려우므로 각 IoT 노드들의 연산 능력을 효율적으로 사용하지 못한다. 또한, 하둡의 기술적 단점을 보완할 수 있는 다양한 분산 시스템 기법이 디자인 되고 있지만 실시간 데이터 처리를 위한 시스템에 적용하기에는 많은 제약이 있다. 이에 아파치 스파크(Apache Spark)[2], 스톰(Strom)과 같은 계층적 데이터 스트림(Stream) 처리에 적합한 플랫폼(Platform)들이 개발되고 있다. 하지만 실질적으로 데이터 처리를 위한 알고리즘의 연구는 부족한 실정임. 따라서 계층적 데이터 스트림 데이터 처리 방법에 대한 비교 분석 연구 개발이 필수적으로 요구되었다.

이에 따라 다양한 데이터 분석처리 알고리즘들이 연구 개발되어 데이터 처리에 적용되어 왔으나, 데이터의 정보 흐름과 타입에 초점을 맞춘 것이 아니라 예상분석 및 획일화된 알고리즘을 통해서 분석되기 때문에 해당 플랫폼이 주로 지향하는 데이터 형식에 맞지 않으면 성능저하를 수반하며 사용에 제약이 많은 문제점이 있다. 또한, 기존의 플랫폼에서는 제한된 알고리즘으로 모든 데이터를 처리, 분석하는 기능을 제공하고 있음. 따라서 다양한 데이터 타입에 기반을 두어 처리엔진 변경에 따른 유동성 보장 및 개별 데이터 목적에 따른 알고리즘 선택을 지원하기 위한 기술 연구가 필요하다.

본 논문에서는 IoT 환경에서 실시간 반응성 향상을 목표로 오픈소스 기반 스트림 데이터 처리 방법에 대한 비교 분석과 Fast-reaction을 위한 데이터 처리 도구 비교 분석 연구를 하는 것으로, 대용량 데이터 처리 및 분산 협업을 지원하는 기계 학습 도구인 MLlib와 아파치 머하웃(Apache Mahout), R, MATLAB의 비교 분석 연구를 통해 Edge computing 환경에서 데이터 처리 속도를 향상시키고, 그 결과를

바탕으로 Fast-Reaction을 위한 최적화된 플랫폼을 선정한다. 이를 증명하기 위해 벤치마크를 통해서 연구된 플랫폼의 스트림 분석 처리를 평가하여 계층적 데이터 스트림 처리 환경에 적합한 FastData 분석 플랫폼을 도출하였다.

II. Related work

2.1 Spark Streaming

스파크 스트리밍(Spark Streaming)은 다양한 데이터 소스로부터 데이터를 받아 실시간 스트리밍을 처리할 수 있도록 해준다는 점에서 스톰과 비슷하다. 그러나 스파크 스트리밍은 스파크 API를 확장해 내장된 머신러닝 알고리즘(Machine learning)을 수행하거나 추가 분석을 할 수도 있다. 또한 클렌징을 통해 데이터를 저장시키고 다시 SQL형식이나 머신 러닝 혹은 일반 스파크 API를 이용해 추가적인 분석을 유기적으로 할 수 있는 장점이 있다. 또한, 트위터(Twitter)에서는 거의 모든 기능을 OpenAPI로 제공하며 API 문서화가 잘되어 있으며, 각 언어별로 상당수의 트위터 라이브러리들이 제공되고 있기 때문에 편리하게 사용할 수 있다.

1.2 Micro blogging

마이크로 블로깅(Micro blogging)으로 큰 인기를 얻고 있는 트위터는 거의 모든 기능을 OpenAPI로 제공하고 있다. 트위터에서 제공하는 API로는 크게 REST와 Streaming 그리고 Search API가 있다. REST API는 트위터의 데이터에 접근할 수 있게 해주는 기능을 제공하며, Streaming API는 트위터 메시지를 실시간으로 사용이 가능하게 해주는 API이다. 마지막으로 Search API는 트위터 타임라인 글의 검색기능을 제공한다.

기존 트위터 사이트는 사용자에게 UX적인 편리함을 제공하지 않는다. 따라서, 대부분의 유저들은 Open API를 이용하여 다른 웹 사이트나 어플리케이션을 통해 간편하게 자신의 편의에 맞추어 트위터를 많이 사용하고 있는 추세이다.

트위터에서 제공하는 트위터 API는 트위터 API KEY를 발급받아야 사용할 수 있다. 트위터의 Open API를 사용하려면 인증과정이 필요한데 트위터의 인증 방식은 OAuth 2.0 기술표준을 따름. OAuth 2.0 인증에 필요한 외부 정보는 현재 인증을 받는 대상을 확인하기 위한 키(Key)와 비밀번호(Secret)이다.

III. The Proposed Scheme

3.1 아파치 스파크 클러스터 시스템 구성

아파치 스파크 시스템은 3개의 클러스터 매니저(Standalone, Apache Mesos, Hadoop Yarn)를 지원하며, 본 논문에서는 3개의 클러스터 매니저 중 대표적인 Standalone 모드를 사용하여 아파치 스파크 클러스터 시스템을 구성하였다. 전체 노드의 구성은 3개의 노드(Master node 1개, Wroker node 2개)로 이루어진다. 마스터 노드의 성능은 4개의 CPU core와 8GB Memory이며, 워커 노드의 성능은 4개의 CPU core와 14GB Memory이다.

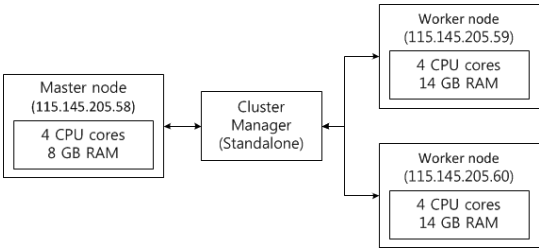

Fig. 1. 아파치 스파크 클러스터 시스템 구조

트위터 데이터를 아파치 스파크 스트리밍을 이용하여 실시간으로 분석하였다. 대표적인 머신러닝 기법인 K-means 기법을 이용하여 군집 모델을 생성하였다. 아파치 스파크 스트리밍에서 트위터 데이터를 실시간으로 받고 K-means 머신러닝 기법으로 생성된 모델을 이용하여 트위터 데이터를 언어별로 분류하였다. 데이터브릭스(Databricks)에서 제공하는 오픈소스를 이용하였으며, 소스코드를 수정하여 K-means 머신러닝 기법의 Training 시간과 Test 시간을 확인하였다.

3.1.1 K-means

클러스터링(Clustering)은 Unsupervised Learning의 일종으로, label 데이터 없이 주어 진 데이터들을 군집 별로 분류하였다. K-means 클러스터링 기법은 k개의 클러스터로 데이터를 나누는 방법이다. 중심 k에 대해서 모든 데이터와의 거리를 구하고, 각 데이터를 가장 가까운 클러스터에 포함되도록 한다. 각각의 클러스터에 포함된 데이터들의 중심을 구하고 클러스터의 중심 k를 이동시킨다. 이 과정을 반복하여 가장 근접한 데이터를 포함하는 클러스터를 구성하는 머신러닝 기법이다.

3.1.2 트위터 데이터 수집

K-means 머신러닝으로 군집 모델을 생성하기 위해 Collect 클래스를 이용하여 100,000개의 트위터 데이터를 수집하였다.

```
Master node : hduser$ ./bin/spark-submit --class "com.databricks.apps.twitter_classifier.Collect" --master spark://115.145.205.58:7077 etri/twitter_kmeans/target/scala-2.10/test1_jar output 100000 1 1 --consumerKey $CONSUMER_KEY --consumerSecret $CONSUMER_SECRET --accessToken ACCESS_TOKEN --accessTokenSecret ACCESSTOKEN_SECRET
```

```
Get number of RDD : 21
Save the outputRDD - File name : 1453274277000
MapPartitionsRDD[13] at foreachRDD at Collect.scala:41
Total get number of RDD : 21
Get number of RDD : 35
Save the outputRDD - File name : 1453274278000
MapPartitionsRDD[20] at foreachRDD at Collect.scala:41
Total get number of RDD : 56
Get number of RDD : 36
Save the outputRDD - File name : 1453274279000
MapPartitionsRDD[27] at foreachRDD at Collect.scala:41
Total get number of RDD : 92
Get number of RDD : 36
Save the outputRDD - File name : 1453274280000
MapPartitionsRDD[34] at foreachRDD at Collect.scala:41
Total get number of RDD : 128
```

Fig. 2. 트위터 데이터 수집

3.1.3 K-means 모델 생성

수집한 트위터 데이터를 이용하여 트위터 데이터를 25개의 클러스터로 나누는 모델을 생성하였다. 모델 생성 후 100개의 트위터 데이터로 모델을 테스트하였다.

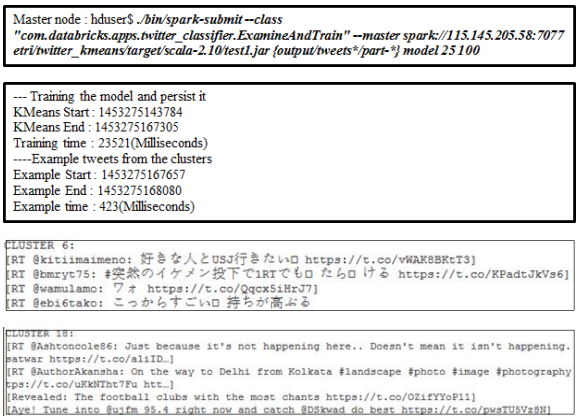

Fig. 3. K-means 모델 생성

3.1.4 실시간 트위터 데이터 분석

아파치 스파크 스트리밍을 통해 트위터 데이터를 실시간으로 받아 K-means 머신러닝 기법으로 생성한 모델을 이용하여 트위터 데이터를 분류하였다. 클러스터 별로 분류된 트위터 데이터 중 지정한 클러스터에 포함되는 트위터 데이터를 확인하였다.

Fig. 4. 트위터 데이터 분석

IV. Conclusions

본 연구에서는 Edge computing에서 군집 지능화를 위한 실시간 반응성 향상을 목표로 오픈소스 기반 스트림 데이터 처리 방법에 대한 비교 분석과 Fast-reaction을 위한 데이터 처리 도구 비교 분석 연구하는 것으로, 오픈 소프트웨어 아파치 스파크를 기반으로 다양한 스트림 데이터 분석 방법(MLlib, R, Mahout, MATLAB 등)들에 대한 비교 연구를 수행하였다.

최근 빅 데이터와 스트리밍 서비스의 발전으로 실시간 스트리밍 데이터 분석 기능을 제공하는 아파치 스파크에 대한 관심이 높아지고 있다. 그러나 기존의 아파치 스파크에 대한 인터넷 서적 등의 자료들은 초기 버전에 대한 자료들이며 현재 제공하고 있는 최신 버전에 대한 자료가 미비한 실정이다. 향후 Edge computing 환경 하에서의 아파치 스파크 기반 데이터 처리 환경 구축 연구를 통해 최신 버전의 아파치 스파크 환경을 구축하고 사용을 할 수 있도록 방향을 제시한다. 또한, 아파치 스파크와 연동되는 스트림 분석 방법 연구에서 FastData인 트위터 데이터를 아파치 스파크 스트리밍을 이용하여 실시간으로 분석하는 사례를 제시하여 아파치 스파크를 빅데이터 분석의 핵심 기술로서 IoT 및 Edge Computing 환경에 적용 시킬 수 있도록 방향을 제시할 것이다.

마지막으로 다양한 스트림 데이터 분석을 위한 FastData 분석 방법을 아파치 스파크와 연동하여 Low Layer부터 High Layer 노드에 적용할 수 있는 메모리 기반의 처리 방식으로 다양한 작업을 빠르고 효율적으로 처리하는 연구를 수행할 것임. 이를 위해 2단계 연구에서는 분산 협업 처리 방법과 기계 학습 도구 비교 분석 연구를 바탕으로 군집 노드의 상황 패턴 인식을 위한 SVM/PCA 기반 학습 알고리즘을 연구를 진행하고자 한다. 학습 알고리즘의 목적은 학습모델과 예측 알고리즘을 만드는 것으로 주어진 학습데이터(Training set)로부터 최적의 학습모델을 생성하고, 해당 모델을 통해 대용량 데이터를 분석, 귀납 추론 등의 결과를 도출하는 방법을 일컫는다. 학습 데이터의 개수에 비례하여 학습 속도와 메모리 용량이 증가하기 때문에 대용량 데이터를 학습하는 데에 기술적인 어려움이 있다. 따라서 분석 및 추론을 위하여 수집된 데이터를 기반으로 상황을 학습하고 예측하여 가장 정확한 서비스를 제공하는데 필요한 SVM/PCA를 포함한 여러 기계 학습 알고리즘을 비교 평가하고 IoT 노드에 분산 및 경량화 시키는 연구가 필요할 것으로 예상된다.

ACKNOWLEDGEMENT

본 연구는 과학기술정보통신부 및 정보통신기술진흥센터의 정보통신·방송연구 개발 사업(No. 2016-0-00133, 초연결 IoT 노드의 군집 지능화를 통한 Edge Computing 핵심 기술 연구), SW중심대학지원사업(2015-0-00914), 한국연구재단 기초연구사업(No.2016R1A6A3A11931385, 실시간 공공안전 서비스를 위한 소프트웨어 정의 무선 센서 네트워크 핵심기술 연구, 2017R1A2B2009095, 실시간 스트림 데이터 처리 및 Multi-connectivity를 지원하는 SDN 기반 WSN 핵심 기술 연구), BK21PLUS 사업의 일환으로 수행되었음.

REFERENCES

[1] Weisong Shi, Schahram Dustdar, “The Promise of Edge Computing”, Computer, Vol. 49, No. 5, pp. 78-81, May 2016

[2] "5Apache Hadoop", [online] Available : http://hadoop.apache.org/.

[3] K Shvachko, H Kuang, S Radia, R Chansler, “The Haddop Distributed File System”, D.H.Ballard, ”Computer Vision,” Prentice-Hall, pp.76-79, 1991.

[4] Jeffrey Dean, Sanjay Ghemawat, “MapReduce: simplified data processing on large clusters”, Communications of the ACM, Vol. 51, No. 1, pp. 107-113, Jan. 2008