

IoT 노드의 데이터 처리를 위한 분산 프레임워크

김민우^{O*}, 이태호^{*}, 이병준^{*}, 김경태^{*}, 윤희용^{**}

^{O*} 상균관대학교 정보통신대학 전자전기컴퓨터공학과

^{**} 상균관대학교 소프트웨어대학 소프트웨어학과

e-mail: {kimmw95, leetaeho, byungjun}@skku.edu^{O*}, kyungtaekim76@gmail.com^{*}, youn7147@skku.edu^{**}

Distributed Framework for Data Processing of IoT Node

Min-Woo Kim^{O*}, Tae-Ho Lee^{*}, Byung-Jun Lee^{*}, Kyung-Tae Kim^{*}, Hee-Yong Youn^{**}

^{O*}Dept. of Electrical and Computer Engineering, Sungkyunkwan University

^{**}Dept. of Software, Sungkyunkwan University

● 요약 ●

분산 컴퓨팅 환경에서 사용되어지는 빅 데이터 파일 시스템은 IoT(Internet of Things) 노드에서 처리해야할 데이터 탐색 시 모든 저장장치를 탐색하기 때문에 속도가 느리며 트래픽으로 인한 오버헤드가 발생할 수 있다. 본 연구에서는 IoT 노드의 분산 컴퓨팅 환경에서 빅 데이터를 좀 더 효율적으로 처리하고 빠른 검색을 위해 머신 러닝 기법을 이용한 분산 프레임워크를 제안하며 IoT 노드에서의 데이터 처리를 위해 다른 저장 장치로의 불필요한 액세스를 사전에 방지하여 빠르고 정확한 연산 결과를 도출하여 효율성을 향상 시키고자 한다.

키워드: 프레임워크(framework), 분산 처리(distributed processing), IoT 환경(IoT environment)

I. Introduction

분산처리 시스템에서의 수행 작업 방식은 기존의 단일 데이터처리 시스템 방식과는 다르게 선점의 개념으로 오버 헤드가 발생하게 된다. 이를 해결하기 위해 본 연구에서는 효율적인 빅 데이터 처리를 위한 프레임 워크를 연구하였다. 기존의 머신 러닝 기법의 스케줄러 알고리즘들의 특성과 복잡도 등을 기준으로 하여 프레임워크 개발 연구를 진행하였다. 이를 통해 머신 러닝 기법을 프레임워크에 적용하여 트래픽을 줄임으로써 IoT(Internet of Things) 노드간의 데이터 처리 효율을 높이는 것을 목표로 한다. 2장에서는 본 연구에서 프레임 워크를 개발하기 위한 기존의 관련 머신 러닝 기법에 대해 서술하였고 [1][2] 3장에서는 제안 하고자 하는 프레임워크의 초기 모델 대한 설명을 제시 한다. 4장에서는 제안된 프레임워크의 초기 모델 대한 피드백과 결론을 서술하였다.

시스템의 높은 가용성을 얻기 위해 데이터 전송 에러에 대한 빠른 복구 방법, 이처럼 문제가 발생했을 때 구성을 고려하여 전면 장애가 발생하지 않기 위한 연구가 선행되어야 한다. 또한 성능 부분에서 데이터 검색 및 처리 시간, 신뢰성 정확성 등이 개발의 주요 사항들이다 [4]. 본 연구에서는 다음의 Table 1의 머신 러닝 기법을 적용하여 IoT 노드의 분산 시스템 환경에서의 프레임워크 초기 모델에 대한 연구를 진행하였다.

Table 1. 머신 러닝 기법

머신러닝 기법	특성	복잡도
GP (Genetic Programming)	Information processing tasks	보통
DT (Decision Tree)	Link quality estimation	낮음
NNs (Nearest Neighbor search)	Dynamic fault detection model	높음
RL (Reinforcement Learning)	QoS task scheduler for adaptive multimedia sensor networks	낮음

II. Preliminaries

1. Related works

1.1 주요 개발 사항

본 연구의 목표로 IoT 노드에서 데이터 처리를 위한 분산 시스템의 처리 속도 및 효율성을 높이기 위한 최적화가 필요하다[3]. 분산

III. The Proposed Scheme

1. 분산 처리 프레임워크

위 2번 섹션에 설명된 4가지 알고리즘을 적용시켜 본 논문에서 제안하는 IoT 환경 기반 분산 처리 프레임 워크 초기 모델에 대해 설명한다. 먼저 GP(Genetic Programming)는 트리 형식의 문제를 표현하며 버그를 수정하기 위한 분석 방법과 결합된다. 또한 트리에서 싱글 노드처럼 표현되고 차후 실행 경로에 따라 지역화(localized)되며 이를 통해 데이터들의 처리를 수행한다. DT(Decision Tree)는 가치 판단에 의해 결정을 하며 추진된 결정에 또 다른 선택지로 나뉘고 그 결정에 의한 각종 문제들이 생성된다. 이처럼 결정에 대한 비교를 할 수 있어 시스템 추진에 유용하게 활용된다. NNs(Nearest Neighbor search)는 최적화 알고리즘이며 기본적인 분류 규칙이다. 요청에 따라 응답으로 데이터 중 비슷한 데이터를 고르지만 그렇지 못한 데이터에 대한 탐지 모델로 사용되었다. RL(Reinforcement Learning)은 환경에 대한 반응을 기반으로 목표에 얼마나 빠르게 효율적으로 찾아가는지에 대한 알고리즘으로 본 논문에서는 IoT 환경에서 노드들의 QoS(Quality of Service)에 대한 작업 스케줄러로 사용되었다. 이를 통해 통합 분석 시스템을 연구하여 분산 시스템 모델들의 구조분석, 관리 및 확장 등 노드 간 데이터 전송 시 데이터 오류, 변조, 손실을 방지할 수 있는 시스템 구축을 목표로 한다.

IV. Conclusions

본 논문에서는 기존의 머신 러닝 기법을 적용하여 분산 프레임워크의 초기 모델을 연구하였다. 향후 연구로는 분산 IoT 노드의 안정성, 네트워크 복잡도와 트래픽 감소에 대한 효율성 등을 분석하고 노드 간 모델들에서의 데이터 무결성, 시스템 알고리즘의 개선 및 최적화를 수행할 계획이다.

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), 삼성전자, BK21PLUS 사업의 일환으로 수행되었음.

REFERENCES

- [1] Su Nguyen, Yi Mei, Mengjie Zhang, "Genetic programming for production scheduling: a survey with a unified framework", *Complex & Intelligent Systems*, Volume 3, Issue 1, pp 41-66, March 2017
- [2] Ling Shao, Li Liu, Xuelong Li, "Feature Learning for Image Classification Via Multiobjective Genetic Programming", *IEEE Transactions on Neural Networks and Learning Systems*, Volume. 25, Issue. 7, pp.1359-1371, July 2014
- [3] Al-Fuqaha A, Guizani M, Mohammadi M, Aledhari M, Ayyash M, "Internet of things: a survey on Enabling technologies, protocols, and application", *IEEE Commun Surveys Tutorials*, pp.2347-2379, 2015.
- [4] Mohammad Abu Alsheikh, Shaowei Lin, Dusit Niyato, Hwee-Pink Tan, "Machine Learning in Wireless Sensor Networks: Algorithms, Strategies, and Applications", *IEEE Communications Surveys & Tutorials*, Volume. 16, Issue. 4, pp.1996-2018, 2014