

# Edge Computing 환경에서 트리 구조를 이용한 인덱스 관리

유승언<sup>0</sup>, 김세준\*, 이병준\*, 김경태\*, 윤희용\*\*

<sup>0</sup>성균관대학교 정보통신대학 전자전기컴퓨터공학과

\*\*성균관대학교 소프트웨어대학 소프트웨어학과

e-mail: {seyoo90, ksj105, byungjun}@skku.edu<sup>0</sup>, kyungtaekim76@gmail.com\*, youn7147@skku.edu\*\*

## Index Management Using Tree Structure in Edge Computing Environment

Seung-Eon Yoo<sup>0</sup>, Se-Jun Kim\*, Byung-Jun Lee\*, Kyung-Tae Kim\*, Hee-Yong Youn\*\*

<sup>0</sup>Dept. of Electrical and Computer Engineering, Sungkyunkwan University

\*\*Dept. of Software, Sungkyunkwan University

### ● 요약 ●

Edge Computing은 분담을 통해 네트워크의 부담을 줄일 수 있는 IoT 네트워크에 적합한 방법으로, 데이터를 전송하고 받는 과정에서 네트워크의 대역폭을 사용하는 대신 서로 연결된 노드들이 협력해서 데이터를 처리하고, 네트워크 말단에서의 데이터 처리가 허용되어 데이터 센터의 부담을 줄일 수 있다. 트리구조는 데이터 구조의 하나로, 데이터 항목의 한 묶음을 세그먼트를 나뉘가지처럼 연결한 것을 의미하여 분산된 데이터를 군집할 수 있다. 본 논문에서는 Edge Computing 환경에서 트리 구조를 이용하여 인덱스를 관리하는 모델을 알아보기 위해 이진 탐색 트리 중 AVL tree와 Paged Binary tree에 대해 서술하였다.

**키워드:** 인덱스 관리(Index management), 이진 탐색 트리(Binary search tree), AVL tree, Paged Binary tree

### I. Introduction

IoT는 성능이 제한되어 있는 기기들을 인터넷을 통하여 연결하여, 다양한 휴대 기기가 통신, 사업 등 여러 분야에 사용되고 있다. 이에 따라 데이터의 양이 급격히 늘어나면서 Edge Computing 환경에 IoT의 노드들이 포함되고 있다. Edge Computing은 분담을 통해 네트워크의 부담을 줄일 수 있는 IoT 네트워크에 적합한 방법으로, 서로 연결된 노드들이 협력해서 데이터를 처리하고, 네트워크 말단에서의 데이터 처리가 허용되어 데이터 센터의 부담을 줄일 수 있다[1].

트리구조는 데이터 구조의 하나로, 데이터 항목의 한 묶음을 세그먼트라고 하는데, 이 세그먼트 사이를 나뉘가지처럼 연결한 것을 의미한다. 상위 세그먼트는 하나 이상의 하위 세그먼트를 가지고 있으나, 하위 세그먼트는 반드시 한 상위 세그먼트밖에 가질 수 없다.

본 논문에서는 Edge Computing 환경에서 트리 구조를 이용하여 인덱스를 관리하는 모델을 알아보기 위해 이진 탐색 트리 중 AVL tree와 Paged Binary tree에 대해 서술하였다.

### II. AVL 트리

AVL트리[1]는 트리를 구성하는 어떠한 노드의 좌측과 우측 subtree의 높이 차이가 1인 트리이며, 균형 잡힌 트리라고도 부른다.

이진 탐색 트리에서 비대칭적 구조에 의한 탐색 효율의 저하를 방지하기 위해서는 트리가 비대칭적으로 형성되는 것을 방지해야 하는데 이를 균형 잡힌 트리(Height-Balanced tree)라고 한다. AVL 트리의 이러한 속성 때문에 AVL 트리를 HB 트리라고 부르기도 한다.

[그림 1]은 Height-Balanced tree의 두 가지 예시를 나타낸다.

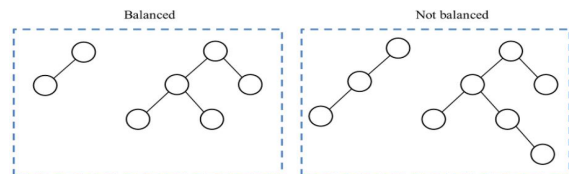


Fig. 1. Height-Balanced tree의 예

### III. Paged Binary tree

AVL 트리는 트리가 비대칭적으로 생성되는 문제점을 해결하였지만, 이진 탐색 자체가 갖고 있는 문제점은 해결할 수 없다. Paged Binary tree[2]는 이진 탐색 자체가 갖고 있는 문제점을 해결하기

위해 고안된 tree 구조로서, 이진트리를 페이지 단위로 분리하여 하드디스크상의 연속된 공간에 저장한다[2]. [그림 2]는 한 페이지에 7개의 노드를 저장하도록 구현된 Paged Binary tree의 한 예시이다.

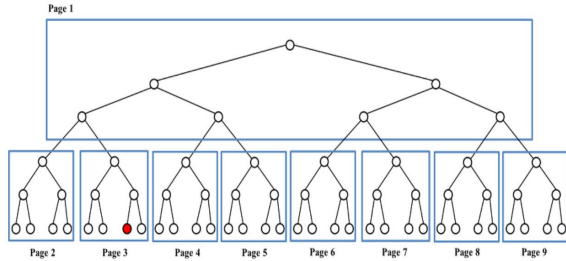


Fig. 2. Paged Binary tree의 예

Paged Binary tree를 이용하여 이진 탐색을 수행할 때, 위 그림과 같이 Page Binary tree에서는 두 번의 seek 연산만으로 원하는 데이터를 읽을 수 있다. 예를 들어, 위 그림에서 붉은색으로 칠해진 노드의 데이터를 하드디스크에서 읽는다고 가정하면 다음의 과정을 통해 데이터를 읽을 수 있다.

- 1) 하드디스크에서 Page1을 읽은 다음, Page1로부터 생성된 이진 트리에서 이진 탐색을 수행한다.(1회의 seek 연산 발생)
- 2) 이진 탐색의 결과 붉은색으로 칠해진 노드의 데이터가 Page3에 저장되어 있다는 결과를 얻게 되고, 하드디스크에서 Page3을 읽는다.(1회의 seek 연산 발생)

#### IV. Conclusions

본 논문에서는 Edge Computing 환경에서 트리 구조를 이용하여 인덱스를 관리하는 모델을 알아보기 위해 이진 탐색 트리의 한 종류인 AVL트리와 Paged Binary tree에 대해 서술하였다. 향후 연구로는 SDN환경에서 분산 컨트롤러에 각 트리를 적용 및 테스트하여 다른 트리 모델과 비교하여 테스트 베드 구축 및 성능평가를 목표로 한다.

#### 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] Xiang Sun, Nirwan Ansari, "EdgeIoT: Mobile Edge Computing for the Internet of Things", IEEE Communications Magazine, Vol. 54, No. 12, pp. 22-29, 16 Dec, 2016
- [2] Nathan G. Bronson, Jared Casper, Hassan Chafi, Kunle Olukotun, "A practical concurrent binary search tree", Proceedings of the 15th ACM SIGPLAN Symposium on Principles and Practice of Parallel Programming, Vol. 45, No. 5, pp. 257-268, May. 2010.