이학학사학위 논문

**PyHadoop을 이용한 분산처리 k-means알고리즘**

Distributed treatment k-means algorithm using pyhadoop

2022. 06.

박지훈

인하대학교 물리학과

이학학사학위 논문

**PyHadoop을 이용한 분산처리** **k-means알고리즘**

Distributed treatment k-means algorithm using pyhadoop

2022. 6.

박지훈

인하대학교 물리학과

지도교수: 이재우 (인)

논문을 이학학사 학위 논문으로 제출함.

이 논문을 박지훈의 이학학사 학위논문으로 인정함.

202 년 월 일

논문사정위원장 (인)

논문사정위원: (인)

논문사정위원 (인)

논문사정위원 (인)

**초 록**

분산처리 프레임워크 Hadoop을 이용하여 Hadoop streaming을 이용해 python으로 분산처리 알고리즘을 구현하였다. 기존의 k-means알고리즘을 단일 PC에서 작동하고 분산처리 k-means알고리즘을 4대의 PC로 이루어진 분산처리 환경에서 작동하여 데이터 사이즈 별 작동시간을 비교하였다. 100MB, 500MB , 1GB 세가지 데이터 사이즈에서 모두 단일 PC의 k-means 알고리즘이 높은 성능을 보였다. 분산 처리 모델은 데이터 사이즈 변화에 대해 일반 모델 보다 둔감하게 반응하는 것을 알 수 있었다. 이를 통해 데이터 사이즈가 작은 경우(1GB미만) 분산처리 모델보다 단일 PC 모델이 더 높은 성능을 띤다는 것과 데이터 사이즈가 커질수록 분산처리 모델의 성능이 점차 우수해지는 것을 확인하였다.

**목 차**

제 1 장 서론

제 2 장 이론

2-1. k-means알고리즘

2-2. Hadoop Distributed File System

2-3. MapReduce

제 3 장 시뮬레이션

3-1. 계발 환경 설정

3-2. 분산처리 실행시간 측정

제 4 장 결론

참고문헌

**제 1 장 서론**

4차 산업 혁명의 도래 이후 빅데이터는 사회의 중요한 키워드가 되었다. 지난 10년 간 사회에서 생산되는 데이터의 양은 기하급수적으로 증가하였다. 데이터 사이즈의 변화는 데이터 분석의 질을 높였다. 그에 더해 데이터 사이즈가 커짐에 따라 기존의 통계학보다 빅데이터에 적합한 부석 방법이 연구되어 사회를 보다 잘 분석하고 예측하게 되었다. 데이터 분석 분야는 이전과는 비교도 할 수 없이 사회에 큰 영향을 미치게 된 것이다.

데이터 사이즈의 증가는 부석의 질을 높였지만 컴퓨터의 실행시간이 증가하는 문제를 야기했다. 물론 컴퓨터 하드웨어도 빠른 속도로 발전하지만 데이터 사이즈 변화의 속도를 따라잡지 못한다. 데이터 분석 결과를 기다리는 시간이 증가하는 것은 모든 분야에서 탐탁치 않은 일이지만 특히 자동차 자율주행과 같은 분야에서는 안전 문제로 이어질 수 있다.

기존에는 계산 시간을 줄이기 위해 컴퓨터의 성능을 높였으나 데이터 사이즈가 증가할수록 하드웨어 비용은 기하급수적으로 올라 일반적인 기업체에서 감당할 수 없었으며 그 마저도 컴퓨터의 기능에 한계가 있었다. 이 문제를 해결하기 위해 데이터 분석에 분산처리 방식을 도입하기 시작하였다. 고가의 PC 하나보다 저가의 PC 여러 개가 데이터 분석 분야에서 같은 비용으로 더 좋은 성능을 나타내는 것이 확인되었기 때문이다.

분산처리 환경은 파일 분산 시스템과 하드웨어 자원과 작업을 관리하는 시스템으로 이루어진다. 본 연구에서는 아파치(Apache) 재단의 프리웨어 자바 소프트웨어 프레임워크인 하둡(Hadoop)을 사용한다. 하둡을 이용하면 분산처리 환경 쓰이도록 작성된 데이터 분석 코드를 실행할 수 있다.

다양한 분석 방법 중 하나인 군집분석은 각 데이터의 유사성을 바탕으로 몇 개의 군집으로 나누고 그 군집 간의 차이를 분석하는 것이다. 알고리즘 비지도학습으로, 분석가는 군집의 존재만을 가정할 뿐 그 개수와 각 특성을 알지 못하는 상태로 분석을 시작한다. 때문에 다수의 분석 시행이 불가피하여 그 실행 시간을 단축시킬 필요가 있다.

군집 분석 중 가장 보편적으로 쓰이는 알고리즘은 K-means이다. 본 연구는 분산처리 환경을 구축과 분산처리 환경에 적합한 K-means 알고리즘을 개발하는 것으로 이루어지며 그 실행 시간을 일반적인 K-means 알고리즘의 실행 시간과 비교하는 것을 목표로 한다.

**제 2 장 이론**

**2.1 K-means 알고리즘**

분석가가 데이터에 군집이 존재한다는 가정을 하고 인간의 지도를 일체 개입하지 않은 채 분석하는 방법을 군집 분석이라고 한다. 때문에 데이터나 방법에 따라 결과나 난이도가 판이하다. 만약 군집이 존재하지 않는 무작위성 데이터의 경우 하나의 군집만을 나타내므로 군집화의 효과를 볼 수 없고, 반면에 데이터의 군집이 깔끔하게 나뉘어져 잘 분리된 군집화가 나타날 수도 있다. 근접성, 밀도, 공유된 특성 등 다양한 기준으로 군집화가 가능하지만 가장 보편적으로 쓰이는 것은 중심점을 기반으로 분석하는 K-means 알고리즘이다.

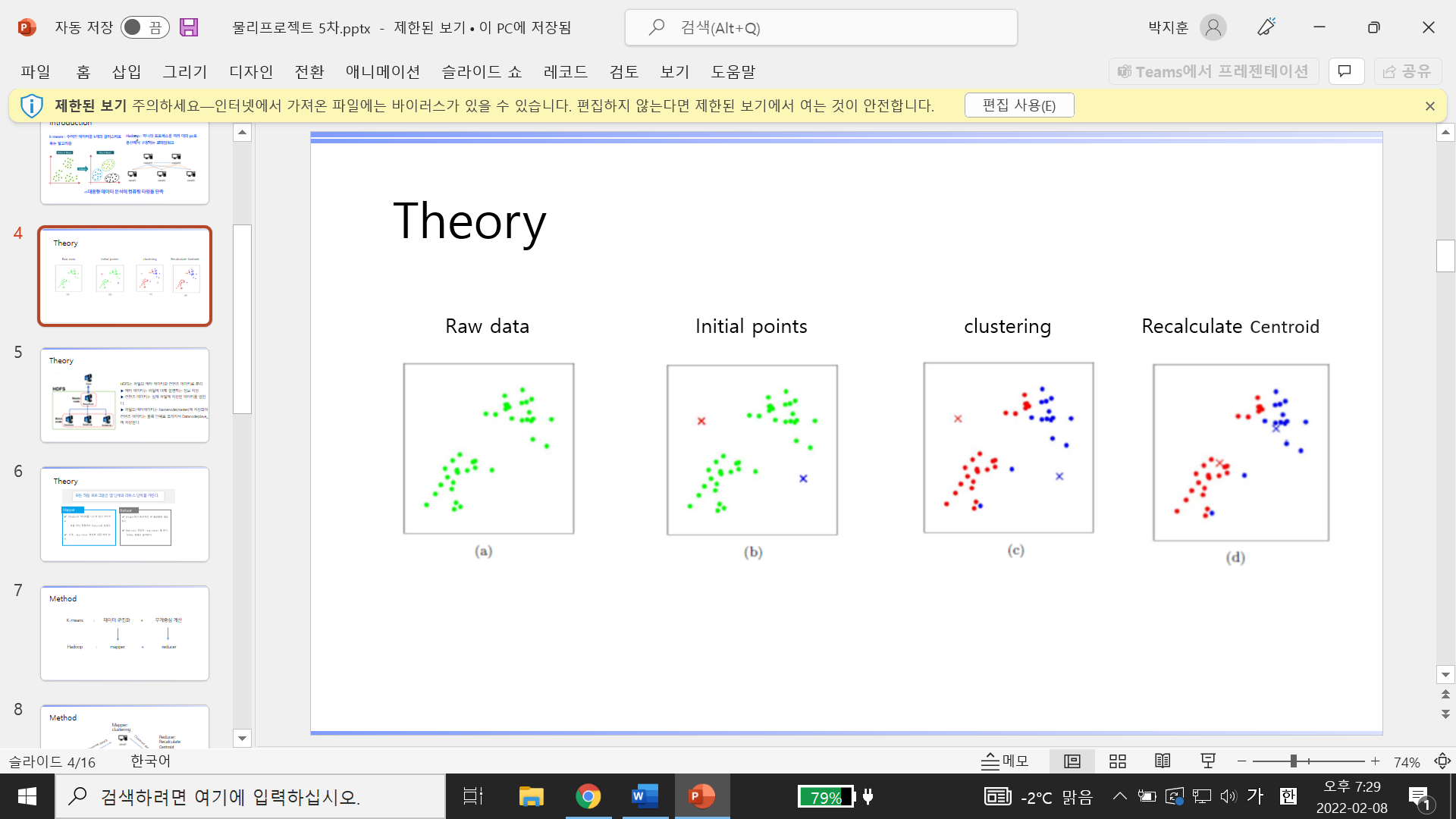


그림1. K-means 알고리즘의 네가지 과정

K-means 알고리즘의‘각 군집의 무게 중심 중에서 가장 가까운 것은 자신이 속한 군집의 중심이다.’라는 가정으로 시작한다. 군집이 지정되지 않은 row data를 변수 공간에 두고 임의의 초기 무게중심을 지정한다. 그 다음 모든 데이터에 대해서 가장 가까운 무게중심을 계산하여 이를 이용해 군집화(clustering)한다. 그리고 각 군집의 무게 중심을 계산하여 무게중심 값을 수정한다(recalculate centroid). Clustering와 recalculate centroid 과정을 반복한다. 이 때 반복 횟수를 지정할 수 있으며 군집의 변화가 임계치 이하일 때 반복을 멈추도록 설정할 수도 있다.

K-means 알고리즘은 분석가가 군집의 개수와 초기 무게중심을 직접 지정해야 하며 그에 따라 판이하게 결과가 달리질 수 있다. 따라서 데이터에 대한 배경 지식이 충분하지 않을수록 여러가지 설정으로 분석을 시행할 필요가 있다. 하지만 K-means 알고리즘은 데이터 사이즈가 증가할수록 실행 시간이 기하급수적으로 늘어나는 특성이 있어 충분한 분석을 시도하기에 어려움이 있다.

**2.2 Hadoop Distributed File System**

분산처리는 하나의 컴퓨팅 프로세스를 여러 대의 PC가 함께 수행하는 것이다. 분산처리에 필요한 데이터는 사이즈가 매우 커 하나의 PC에 저장할 수 없는 경우가 많으며, 가능하더라도 큰 비용을 소모한다. 때문에 여러 대의 PC의 저장공간을 이용해 가상의 저장공간을 만들어 데이터를 저장한다.

아파치 재단은 보다 효율적으로 분산처리 데이터를 저장할 수 있는 시스템인 하둡 분산 파일 시스템(HDFS: Hadoop Distributed File System)을 개발하여 제공한다.

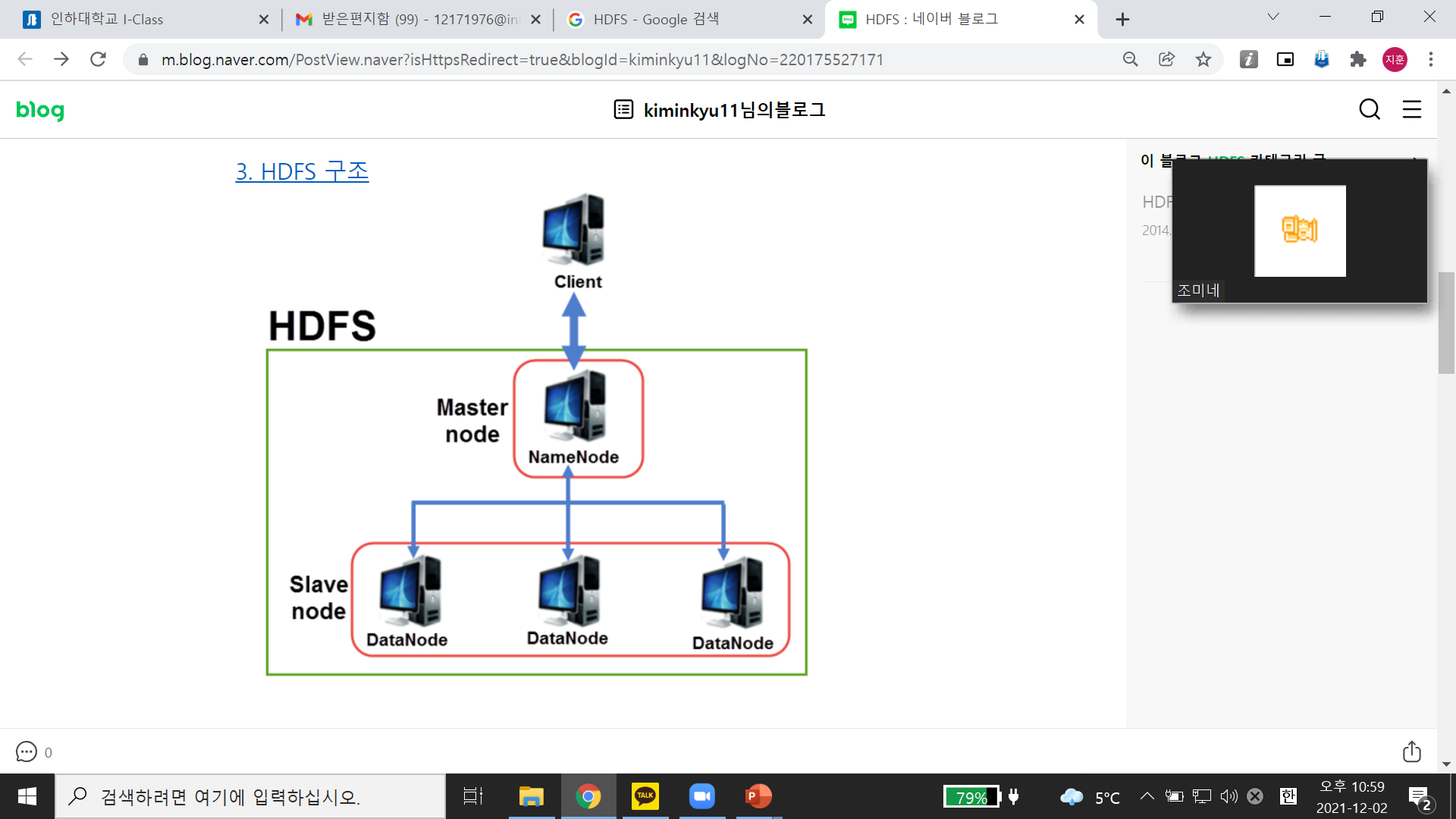


그림2. HDFS 도식

분산처리에 참여하는 PC는 분산된 데이터에 대한 명령어를 실행하는 Data Node, 통합된 데이터에 대한 명령어를 실행하는 Name Node, 모든 프로세스를 관리하는 Client로 구분된다. Client와 Name Node는 각각 한 대이지만 Data Node는 여러 대로, 많은 Data Node를 사용할수록 더 좋은 성능의 분산처리 환경을 구축할 수 있다.

데이터는 데이터의 내용을 담는 컨텐츠 데이터와 데이터에 대한 정보를 담는 메타 데이터로 나뉜다. 이 때, 메타데이터는 Name Node에 저장되고, 컨텐츠 데이터는 분산되어 Data Node에 저장된다. 데이터의 어느 부분을 어느 Data Node가 저장할지 여부는 Client가 관리한다. 때문에 분석가는 Client를 통해 데이터를 다룰 뿐 데이터의 어느 부분이 어느 Data Node에 저장되어 있는지는 확인할 필요 없다.

**2.3 MapReduce**

분산처리 시스템은 입력된 데이터를 Data Node의 개수만큼 나눈다. 그리고 각 Data Node는 분산된 데이터에 대한 분석 명령인 Mapper를 실행한다. 이 때 Mapper의 결과는 항상 key와 value의 쌍으로 이루어진다. Mapper 실행이 끝나면 그 결과를 한데 묶어 Reducer의 입력이 된다. 이때 Reduce는 Name Node에서 실행된다. Name 노드는 그 결과를 모아 합산하여 최종 결과를 출력한다. 이러한 명령어 구조를 MapReduce라고 한다.

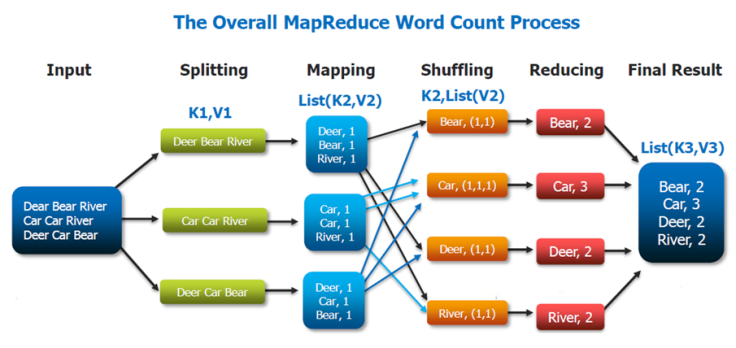


그림3. MapReduce 시스템의 예시

분산처리 분석 코드는 Mapper와 Reducer로 나뉘어져야 하며 그 사이에는 key와 value의 쌍으로 값이 오가야 한다. 본 연구에서는 K-means 알고리즘의 Clustering을 Mapper로 구현하고 recalculate centroid를 Reducer로 구현할 것이다. 각 데이터에 대해서 분산된 Data Node가 Mapper를 실행해서 어느 군집 중심에 가까운지 계산하여 출력하고 그 출력 값을 합산하여 Name Node가 Reducer를 실행하여 새로운 무게중심을 계산하는 것이다.

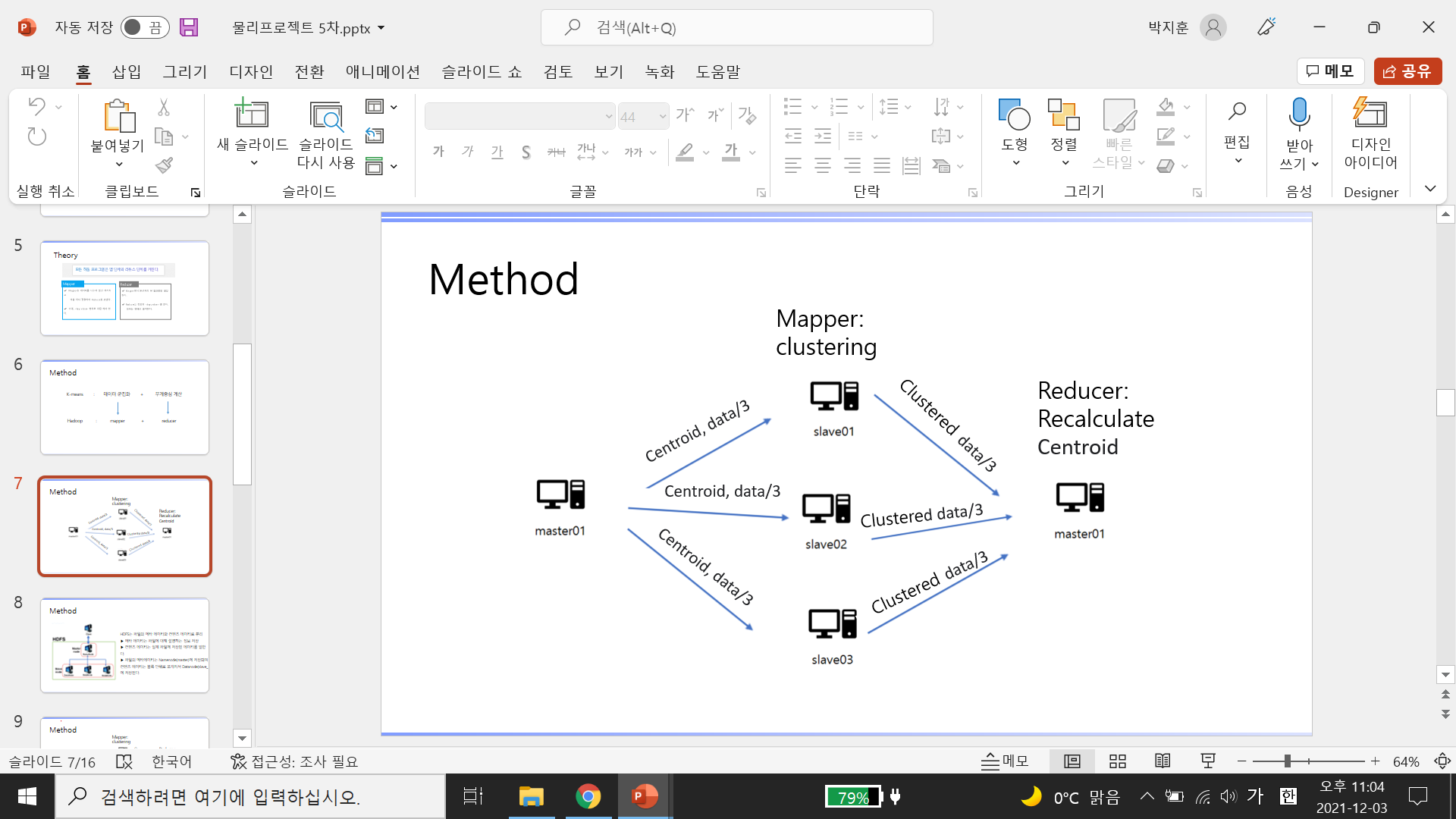


그림4. 분산처리 K-means 알고리즘 모식도

무게중심 값을 저장하는 파일을 따로 두고 Mapper에서 이를 읽어 각 데이터가 어떤 무게중심에 가장 가까운지 계산할 수 있도록 하였다. 그리고 Reducer는 무게중심을 계산하여 파일을 수정하도록 하였다. 두 과정을 반복하고 군집의 변화율이 1% 미만인 경우 반복을 멈추도록 하였다.

**제 3 장 시뮬레이션**

**3.1 계발 환경 설정**

본 연구는 1대의 Name Node와 3대의 Data Node로 진행되었다. 각 PC는 같은 WIFI를 통해 연결하였다.

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **PC 사양** | **Name Node** | **Data Node1** | **Data Node2** | **Data Node3** |
| **CPU** | 11th Gen Intel(R) Core(TM) i5-1135G7 | AMD Ryzen 5 PRO 3500U w | Intel(R) Core(TM) i5-6200U | Intel(R) Core(TM) i5-10210U CPU |
| **RAM** | 8.00GB | 16.0GB | 8.00GB | 8.00GB |
| **가상머신 RAM** | 4GB | 4GB | 4GB | 4GB |

표1. PC 사양

하둡은 프리웨어 자바 소프트웨어 프레임워크이며 여러 대의 PC가 온라인으로 연결되어 하나의 서버를 이루어야 하므로 리눅스 운영체제에서만 작동된다. 본 연구에서는 윈도우 운영체제에서 컴퓨터 가상화 소프트웨어를 이용해 가상의 눅스 환경을 만들어 리눅스 환경을 구축하였다. 리눅스 환경에서 하둡과 자바를 설치하였다.

하둡은 자바로 구성되어 있지만 하둡에서 지원하는 하둡 스트리밍(Hadoop streaming) 기능을 이용하여 python으로 작동이 가능하다. 대부분이 자바 언어로 작동되지만 Mapper와 Reducer의 파이썬 코드 파일을 입력함으로써 mapping, reducing 기능만은 python으로 작동하도록 하는 것이다.

|  |  |
| --- | --- |
| **Software** | **version** |
| **VMware** | 16.0 |
| **Java** | 1.8.0 |
| **Hadoop** | 3.3.1 |
| **Python** | 3.8.3 |

표2. 소프트웨어 버전

|  |  |
| --- | --- |
| **가상머신 환경** |  |
| **리눅스** | CentOS 7 |
| **Hard Disk** | 50GB |
| **RAM** | 4GB |
| **Processor Core** | 2개 |

표3. 가상환경 설정

**3.2 시뮬레이션 결과**

본 연구에서는 분산처리 환경에서 작동하는 K-means 알고리즘과 일반적인 단일 PC 환경에서 작동하는 K-means 알고리즘을 구축여 두 알고리즘의 실행시간을 비교하고자 한다. 단일 PC 알고리즘은 Name Node에서 작동되고 분산 처리 알고리즘은 1대의 Name Node와 3대의 Data Node를 사용하여 작동된다.

실행 시간을 특성하기 위해 사용한 데이터는 반도체 공정 데이터이다. 데이터는 5개의 열로 이루어져 있고 각 열은 연속형 변수를 가진다. 데이터 사이즈에 대한 변화를 확인하기 위해 100MB(1570000lines),500MB(96250000lines), 1GB(51975000lines)에 대해 각각 실행 시간을 측정하였다.

텍스트이(가) 표시된 사진

자동 생성된 설명

그림5. 실행시간 측정을 위한 데이터 예시

|  |  |  |  |
| --- | --- | --- | --- |
|  | **단일 PC** | **분산 처리** | **단일 PC/ 분산 처리** |
| **100MB**  **(1570000lines)** | 9.9837s | 56.83s | 5.69 |
| **500MB**  **(96250000lines)** | 45.8592s | 94.6346s | 2.063 |
| **1GB**  **(51975000lines)** | 84.6575s | 168.7842s | 1.994 |

표4. 단일 PC 실행시간과 분산처리 실행시간 비교

표4에 각 데이터 사이즈에 따른 단일 PC와 분산처리의 실행시간을 비교하고 그 비율을 정리하였다. 모든 경우에서 분산처리 모델이 더 많은 실행시간이 소모되었다. 하지만 데이터 사이즈가 증가할수록 그 차이는 비율이 점차 감소하는 것을 확인할 수 있었다.

**제 4 장 결론**

단일 PC와 분산처리 모델에서의 K-means 알고리즘의 실행시간을 비교하였다. 보든 경우에서 분산처리 모델이 더 많은 실행시간을 소모하였다. 그 원인을 분석하였다.

분산처리 모델은 HDFS 하에서 각 PC 사이에 데이터를 주고받으며 작동된다. 본 연구에서는 각 PC가 무선으로 연결되어 데이터를 주고받는데 시간이 많이 소모되었을 것으로 보인다. 또한 분산처리 모델은 MapReduce 구조를 갖추어야 하기 때문에 제약이 없는 단일 PC의 경우보다 많은 명령어를 포함하는 것이 불가피하다. 여러 대의 PC를 이용해 시간이 단축된 명령어도 있지만 단일 PC에서는 다룰 필요가 없는 명령어가 포함되어 실행시간 증가에 영향을 미친 것으로 보인다.

분산처리 모델은 4대의 PC, 1GB 이하의 데이터에서는 단일 PC에 못 미치는 성능을 나타낸다. 또한 데이터 사이가 큰 경우에서 분산처리 모델의 성능이 올라가는 것을 확인할 수 있었다. 분산 처리 모델은 데이터 사이즈 변화에 대해 일반 모델 보다 둔감하게 반응하는 것을 알 수 있었다. 이를 통해 보다 많은 데이터 사이즈에서 실험하면 분산처리 알고리즘이 더 적은 실행 시간을 소모할 것이라고 예측하였다. 향후 더 많은 PC를 유선 연결하고 더 많은 데이터를 이용해 소모 시간을 측정할 필요가 있다.

**참고문헌**

[1] Dae-Cheol Kang, Jin-Young Choi, Jik-Soo Kim(2018). “Study on performance and stable execution environment of distributed message processing system based on Hadoop”. The Korean Institute of Information Scientists and Engineers

[2] Seoung-Beom Heo, Dae-Cheol Kang, Jin-Young Choi(2019). “Technology Trends of Deep Learning Framework on Hadoop YARN”. The Korean Institute of Information Scientists and Engineers

[3] SungHwan Jeon, HaeJin Chung, YunMook Nah(2017). “Improving Machine Learning Perfomance by Hadoop Mapreduce Tuning”. The Korean Institute of Information Scientists and Engineers

[4] Jun-Young Jeong, Ho-Yon Hwang(2021). “Selection and Evaluation of Vertiports of Urban Air Mobility (UAM) in the Seoul Metropolitan Area using the K-means Algorithm”. Journal of Advanced Navigation Technology.

[5] Ji-Eun Shin, Byung-Ho Jeong, Dong-Hoon Lim(2015), “RHadoop을 이용한 빅데이터 분산 처리 시스템”, Journal of the Korean Data Information Science Society.