한국컴퓨터정보학회 동계학술대회 논문집 제24권 제1호 (2016. 1)

Key개수가 MapReduce 성능에 미치는 영향에 관한 연구

정석준*, 김진홍*, 신동렬*
*성균관대학교 정보통신대학 컴퓨터공학과
e-mail: {sjjeong1225, jinhkm}@gmail.com*, drshin@ece.skku.ac.kr*

A Study on the effect of the number of Key to MapRedue performance

Seok-Jun Jeong*, Jin-Hong Kim*, Dong-Ryeol Shin*
*College of Information and Communication Engineering, Sungkyunkwan University

● 요 약 ●

정보통신기술의 급속한 발전으로 인해 인터넷은 사회 전 분야를 변화시키고 있고 이를 통해 데이터의 양이 증가하면서 의료, 교육, 경영 등 사회 전 분야에서 빅데이터에 관심이 증가하고 있다. 이에 따라 다양한 빅데이터 오픈소스가 생기고 데이터의 크기에 따라 성능을 비교하는 실험이 진행되었다. 본 논문에서는 데이터의 크기가 아니라 데이터를 분류하는 key의 개수에 따라 성능을 비교하고자 한다.

키워드: Hadoop, MapReduce, R, RHadoop

I. 서론

지난 수년간 스마트폰과 같은 스마트기기의 빠른 확산과 함께 SNS 등 소셜미디어가 급성장함에 따라 개인 정보와 소비 패턴, 위치 정보 등이 포함된 가치 있는 데이터가 매순간 엄청난 양으로 생성되고 있으며, M2M(Machine to Machine)과 IoT(Internet of Things) 등이 활성화되면서 인프라 자체도 다량의 데이터를 직접 생성하기 시작했다. 이렇게 새롭게 생성·유통되는 방대한 양의 데이터, 즉 빅데이터의 상당수는 비정형 데이터로서 전체 데이터양의 약 80%에 달하고 있다. 기존 방식으로는 이들 빅데이터 분석에 상당한 비용과 시간이 소요되고 분석이 제한됨에 따라 빅데이터에 맞는 새로운 방식과 기술이 등장하게 되었다. 또한 R언어와 Hadoop의 기능을 융합한 RHadoop과 R Studio를 통한 Web Interface로 빅데이터를 처리할 수 있게 되었다.[1]

II. 관련 연구

2.1 Hadoop

Hadoop은 대량의 자료를 처리 할 수 있는 분산 처리 프로그램을 지원하는 자바 소프트웨어 프레임워크이다 [2]. 분산처리 시스템을 위해 Hadoop은 분산 파일 시스템(HDFS: Hadoop Distributed File System)과 분산 처리 시스템(MapReduce)로 구성되어 있다. HDFS와 MapReduce는 Master / Slave 구조로 구성되어 있으며, HDFS는 NameNode / DataNode로 관리되며 MapReduce는 ResourceManager / NodeManager로 관리된다. HDFS는 NameNode와 DataNode로 구성 되어 있다. NameNode가 파 일의 메타정보를 관리하고 실제 데이터는 여러 대의 DataNode에 분산해서 저장한다 [3-5]. 데이터는 일정 크기의 블록단위로 관리되며 이 블록들을 여러 대의 DataNode에 분산 및 복제해서 저장한다. 분산 및 복제해서 저장함으로써 일부 DataNode에 장애가 발생하더라도 복구가 가능하다.

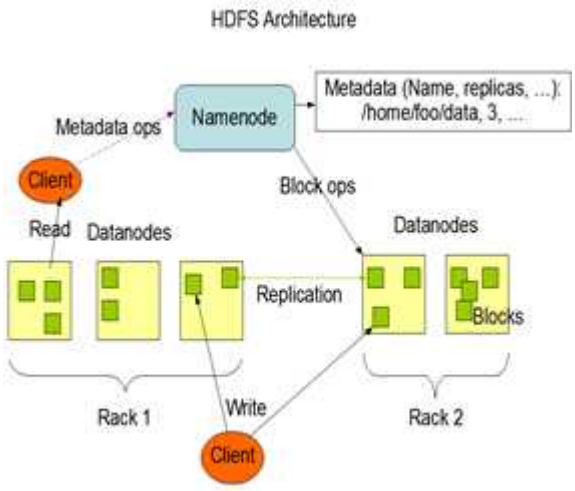

Fig 1. HDFS Architecture

MapReduce는 HDFS에 저장된 데이터를 분산처리하기 위한

Map과 Reduce로 구성되어 있는 프로그래밍 모델이다. Map은 데이터를 Key와 Value의 형태로 구분하는 작업이며 Reduce는 Map으로 인해 Key와 Value로 나눠진 데이터를 Key를 기준으로 Value를 수집하는 작업이다.

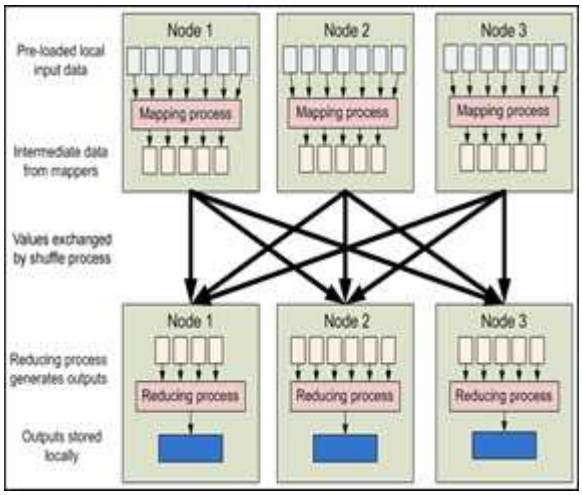

Fig 2. MapReduce Architecture

2.2 R

R은 통계계산과 그래프를 위한 언어이다 [6]. R은 통계소프트웨어 개발과 자료 분석에 널리 사용되고 있으며, 패키지 개발이 용이하여 통계학자들 사이에서 통계 소프트웨어 개발에 많이 쓰이고 있다. 다양한 통계 기법과 수치 해석 기법을 지원하며 사용자가 제작한 패키지를 추가하여 기능을 확장할 수 있다.

2.3 RHadoop

RHadoop[7]은 R언어 환경에서 Hadoop의 기능을 사용해 데이터를 분석하고 관리하기 위한 R Package이다. Hadoop의 HDFS와 MapReduce은 rhdfs와 rmr패키지로 사용이 가능하다

III. 실험 환경

실험환경은 아래와 같은 서버를 사용했다.

OS	Ubuntu Server 14.04
CPU	Intel Xeon CPU E5-2630 v2 2.60GHZ
Memory	32 GB 1600MHz
HDD	1 TB

Hadoop Single Mode로 시스템을 구성했으며 RHadoop으로 MapReduce 성능을 측정하였다.

아래의 R Code를 사용해 실험하였다

```
R Language
library(rhdfs)
hdfs.init()
library(rmr2)
makeword = function(length, size){
words = NULL
for(i in 1:length){
words = paste(words,floor(runif(size)*10),sep = "")
}
for(i in 1:(9-length)){
words = paste(words,1,sep="")
}
return(words)
}
mapper = function(key, val){ keyval(val,1) }
reducer = function(key, val){ keyval(key,sum(val)) }
wordsize = 1000
timeresults = c(1,2,3,4,5,6,7,8)
for(i in 1:5){
wordsize = wordsize*10
seconds = NULL
for(j in 1:8){
fileConn<-file("/home/rhadoop/paper/words.txt")
writeLines(makeword(j,wordsize), fileConn)
close(fileConn)
hdfs.put("/home/rhadoop/paper/words.txt","/user/rhadoop/input/words.txt")
time = proc.time()
results = mapreduce("input/words.txt", input.format="text", map=mapper, reduce=reducer)
time = proc.time() - time
seconds = c(seconds,time[3])
}
timeresults = data.frame(timeresults,seconds)
write.table(timeresults,"/home/rhadoop/paper/result.txt")
}
```

임의의 단어를 만드는 makeword()함수를 구현했으며 wordcount기능을 위해 mapper와 reducer를 만들었다. makeword()함수를 통해 만들어진 임의의 단어들을 words.txt로 hdfs에 저장하고 mapreduce()함수 전후로 시간을 측정했다. 단어의 종류를 10^length개로 총 8개의 케이스로 구분했으며 8개의 케이스를 각각 10,000개부터 10배수로 100,000,000개 까지 실험하였다.

IV. 실험 결과

Table 1에서 가로축에 1 ~ 8의 값은 생성 될 수 있는 key의 10^n개의 개수이고 세로축의 10,000 ~ 100,000,000은 생성되는 단어의 총 개수이다. 10^n개의 key에서 세로축의 개수만큼 데이터를 랜덤으로 만들어 실험을 진행하였다. 데이터의 단위는 second이다.

	1	2	3	4
10,000	25.04	26.17	29.94	35.25
100,000	27.35	27.05	27.19	27.91
1,000,000	25.83	26.95	28.86	41.91
10,000,000	43.03	46.93	64.16	168.76
100,000,000	335.6	245.1	371.57	1408.01
	5	6	7	8
10,000	28.37	28.56	38.51	39.4
100,000	37.05	46.44	46.34	52.31
1,000,000	64.07	159.55	214.78	232.99
10,000,000	258.78	428.94	1374.21	1993.96
100,000,000	2290.74	2659.19	4064.91	13477.45

Table 1. 실험결과

Table 1을 차트로 만들면 Chart 1의 결과가 나온다. Chart 1에서 볼 수 있듯이 데이터의 개수가 1천만개 일 경우에는 key의 개수가 10^7개이면 실행시간이 급격하게 늘어난 것을 확인 할 수 있었으며 데이터의 개수가 1어개일 경우에는 key의 개수가 10^4개부터 실행시간이 급격하게 늘어난 것을 확인 할 수 있었다.

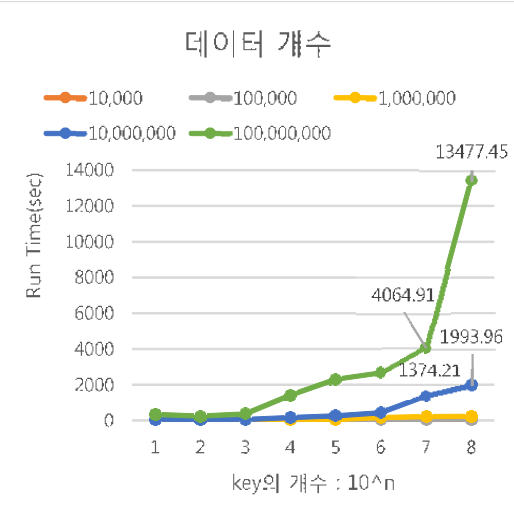

Chart 1. 실험결과

V. 결 론

본 논문에서는 Key의 개수에 따른 MapReduce의 성능비교 실험을 하였다. Table 1과 Chart 1에서 볼 수 있듯이 데이터의 크기가 크더라도 데이터의 분류기준인 key의 개수가 적으면 실행 시간이 적게 걸리는 것을 확인 할 수 있다. 따라서 같은 데이터라도 분류기준을 적게 만들면 실행시간을 단축시킬 수 있는 결론을 얻을 수 있다.

References

[1] S.J Jeong, J.H Kim, D.R Shin. A Study of Cloud Computing Environment for Big Data

[2] https://hadoop.apache.org

[3] F. Chang, J. Dean, S. Ghemawat, W. C. Hsieh, D. A. Wallach, M. Burrows, T. Chandra, A. Fikes, and R. E. Gruber. Bigtable: A distributed storage system for structured data. ACM Trans. Comput. Syst., 26(2):1–26, June 2008.

[4] Pavlo, A., Paulson, E., Rasin, A., Abadi, D.J., DeWitt, D.J., Madden, S., and Stonebraker, M. A comparison of approaches to large-scale data analysis. In Proceedings of the 2009 ACM SIGMOD International Conference (Providence, RI, June 29–July 2). ACM Press, New York, 2009; http://database.cs.brown.edu/projects/mapreduce-vs-dbms/

[5] Pike, R., Dorward, S., Griesemer, R., and Quinlan, S. Interpreting the data: Parallel analysis with Sawzall. Scientific Programming Journal, Special Issue on Grids and Worldwide Computing Programming Models and Infrastructure 13, 4, 227–298.

[6] https://www.r-project.org/about.html

[7] https://github.com/RevolutionAnalytics/RHadoop/wiki

[8] https://www.rstudio.com/