

제한된 메모리 환경에서의 아파치 스파크 성능 비교

송준석*, 김상영*, 이정준^o, 윤희용*

^o성균관대학교 전자전기컴퓨터공학과

e-mail: {alskpo, impsoft, jungjune86, youn7147}@skku.edu^o

A performance comparison for Apache Spark platform on environment of limited memory

Jun-Seok Song*, Sang-Young Kim*, Jung-June Lee^o, Hee-Yong Youn*

^oDept. of Electrical and Computer Engineering, Sungkyunkwan University

● 요약 ●

최근 빅 데이터를 이용한 시스템들이 여러 분야에서 활발히 이용되기 시작하면서 대표적인 빅데이터 저장 및 처리 플랫폼인 하둡(Hadoop)의 기술적 단점을 보완할 수 있는 다양한 분산 시스템 플랫폼이 등장하고 있다. 그 중 아파치 스파크(Apache Spark)는 하둡 플랫폼의 속도저하 단점을 보완하기 위해 인 메모리 처리를 지원하여 대용량 데이터를 효율적으로 처리하는 오픈 소스 분산 데이터 처리 플랫폼이다. 하지만, 아파치 스파크의 작업은 메모리에 의존적이므로 제한된 메모리 환경에서 전체 작업 성능은 급격히 낮아진다. 본 논문에서는 메모리 용량에 따른 아파치 스파크 성능 비교를 통해 아파치 스파크 동작을 위해 필요한 적정 메모리 용량을 확인한다.

키워드: 빅 데이터(big data), 아파치 스파크(apache spark), 인 메모리(In-memory)

I. Introduction

아파치 스파크(Apache Spark)는 대용량의 데이터를 효율적으로 처리하기 위한 오픈 소스 클라우드 플랫폼이다[1]. 아파치 스파크는 Resilient Distributed Dataset(RDD)라 부르는 인 메모리 처리 방식을 지원한다[2]. 아파치 스파크의 인 메모리 기반 데이터 처리 방식은 머신 러닝과 그래프 알고리즘과 같은 반복적인 어플리케이션에 적합하다. 하지만, 아파치 스파크의 작업 실행 시간은 자원 할당 최적화를 위해 필요한 노드 개수와 CPU 속도, 메모리 용량 등에 의존하므로 전체 작업 성능을 예측하기가 어렵다. 본 논문에서는 메모리 용량에 따른 아파치 스파크 성능 비교를 통해 아파치 스파크 동작을 위해 필요한 적정 메모리 용량을 확인한다.

저장장치에 실제로 존재할 필요는 없다. 각 RDD는 서로 다른 노드들에서 RDD의 연산을 가능하게 하기 위한 충분한 정보를 가지고 있다. 이러한 특성으로 인해, 현재 진행 중인 노드의 작업이 실패하더라도 기존의 정보를 이용하여 RDD를 항상 재구성 할 수 있다.

III. Performance Comparison

본 논문에서는 아파치 스파크 성능 비교를 위한 전체 노드의 구성은 그림 1. 과 같다.

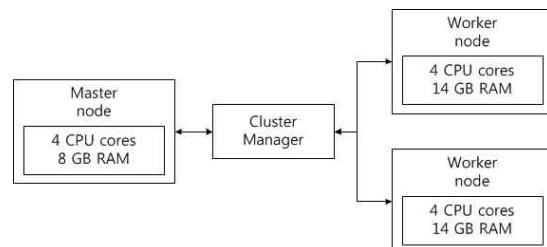


Fig. 1. Simulation Setup

II. Preliminaries

1. Related works

1.1 Resilient Distributed Dataset(RDD)

RDD는 파티션으로 나누어진 변경 불가능한 객체들의 집합이다[3]. 아파치 스파크의 모든 작업은 RDD 생성, RDD 변경, RDD 연산 호출로 이루어진다. RDD는 반드시 안정된 저장장치 안의 데이터나 다른 RDD에서의 결정론적인 실행에 의해 생성되며, RDD는 실제

아파치 스파크 환경 구축은 Ubuntu 환경에서 Java 1.7, Scala 2.10.4, Apache Spark 1.5.1을 이용했으며, Wikipedia dump data와 'WordCount' 어플리케이션으로 실행 시간을 측정했다. 각각의 CPU

core에 1개의 task를 할당하기 위해 Wikipedia dump data의 용량은 512MB로 설정했다.

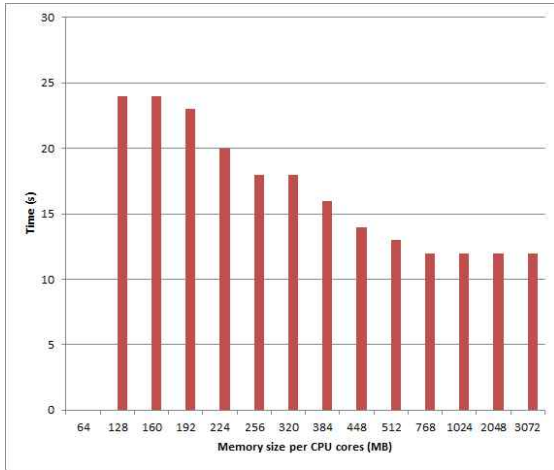


Fig. 2. Simulation Result

메모리 용량에 따른 'WordCount' 작업 실행 시간은 그림 2. 와 같다. CPU core 당 64 MB(총 512MB) 이하 메모리 용량의 환경에서는 'Word Count'가 실행되지 않으며, 메모리 용량이 늘어날수록 실행 시간이 줄어든다. CPU core 당 512 MB(총 4GB) 이상일 경우 실행시간이 12~13초로 일정하다.

IV. Conclusions

본 논문에서는 제한된 메모리 환경에서의 아파치 스파크의 성능을 비교했다. 아파치 스파크 어플리케이션에 사용되는 총 메모리 용량이 입력 데이터의 용량과 같은 경우 어플리케이션은 동작하지 않았으며, 메모리 용량이 증가할수록 어플리케이션 실행 시간이 줄어들다가 CPU core 당 메모리 용량이 입력 데이터 용량보다 커지면 실행 시간이 일정해지는 것을 확인했다. 성능 비교 결과를 통해 인 메모리 기반의 아파치 스파크 성능을 활용하기 위한 적정 CPU core 당 메모리 용량은 어플리케이션의 입력 데이터 용량과 같거나 커야한다는 것을 알 수 있었다.

향후 연구 방향으로 제한된 메모리 환경에서의 아파치 스파크의 실행 시간을 예측하기 위해 메모리 용량에 따른 아파치 스파크 성능을 모델링하는 방안에 대해 연구를 진행할 예정이다.

Acknowledgment

본 연구는 BK21Plus 사업, 한국연구재단 기초연구사업 (2013R1A1A2060398), 삼성전자, 미래창조과학부 및 정보통신기술 연구진흥센터의 정보통신·방송 연구개발사업 (1391105003), 미래부 /정보통신방송연구개발사업의 일환으로 수행하였음.

References

- [1] K. Wang, Khan, M.M.H, "Performance Prediction for Apache Spark Platform," 2015 IEEE 17th International Conference on High Performance Computing and Communications, pp. 166-173, 2015.
- [2] M. Zaharia, M. Chowdhury, MJ Franklin, S. Shenker, I. Stoica, "Spark: cluster computing with working sets," Proceedings of the 2nd USENIX conference on Hot topics in cloud computing, Vol. 10, No. 10, June 2010.
- [3] M. Zaharia, M. Chowdhury, T. Das, A. Dave, J. Ma, M. McCauley, MJ Franklin, S. Shenker, I. Stoica, "Resilient distributed datasets: A fault-tolerant abstraction for in-memory cluster computing," Proceedings of the 9th USENIX conference on Networked Systems Design and Implementation, Vol. 2, No. 2, Apr. 2012.