한국컴퓨터정보학회 동계학술대회 논문집 제24권 제1호 (2016. 1)

Spark 애플리케이션 기반의 성능 분석

정영교*, 이병준*, 조영주O, 윤희용*
O*성균관대학교 정보통신대학
e-mail:{joung0491, byungjun, yjcho1021, youn7147}@skku.eduO*

A Performance Analysis Based on Spark Application

Young Gyo Jung*, Byung-Jun Lee*, Young-Joo ChoO, Hee Yong Youn*
O*Dept. of Electrical and Computer Engineering, Sungkyunkwan University

● 요 약 ●

아파치 스파크는 효율적으로 대용량 데이터를 처리하기 위해 분산 메모리 추상화를 사용하는 오픈 소스 분산 데이터 처리 플랫폼이다. 하지만 아파치 스파크 플랫폼의 특정 작업의 성능은 입력 데이터의 유형과 크기, 디자인 및 알고리즘의 구현 및 컴퓨팅 능력에 따라 메모리 사용량 및 I/O 비용이 크게 달라질 수 있다는 문제점이 있다. 이러한 문제점을 해결하기 위하여 본 논문에서는 아파치 스파크 플랫폼에 대한 높은 정밀도 작업 성능을 예측할 수 있도록 CPU core수의 증가에 따른 WordCount 시뮬레이션을 비교 평가 하였다.

키워드: 스파크(Spark), RDDs(Resilient Distributed Datasets), WordCount

I. Introduction

많은 클라우드 컴퓨팅 플랫폼 중 아파치 스파크[1]는 분산 메모리를 활용하여 대량의 데이터 볼륨의 빠른 처리를 가능하게 RDDs(Resilient Distributed Datasets)[2]의 개념을 사용하는 오픈 소스 클라우드 플랫폼 중 하나이다. 또한 효율적으로 대용량 데이터를 처리하기 위해 분산 메모리 추상화를 사용하는 오픈 소스 분산 데이터 처리 플랫폼이다. 하지만 아파치 스파크 플랫폼의 특정 작업의 성능은 입력 데이터의 유형과 크기, 디자인 및 알고리즘의 구현 및 컴퓨팅 능력에 따라 메모리 사용량 및 I/O 비용이 크게 달라질 수 있다는 문제점이 있다.[3] [4] 이러한 문제점을 해결하기 위하여 본 논문에서는 아파치 스파크 플랫폼에 대한 높은 정밀도 작업 성능을 예측할 수 있도록 CPU core수의 증가에 따른 WordCount 시뮬레이션을 하여 비교 평가 하였다.

II. Preliminaries

1. Related works

1.1 RDDs(Resilient Distributed Datasets)

스파크 내에 저장된 데이터를 RDD라고 하며 변경이 불가능하다. 변경을 하려면 새로운 데이터 셋을 생성해야 한다. RDD는 여러 분산 노드에 걸쳐서 저장되는 변경이 불가능한 데이터의 집합으로 각각의 RDD는 여러 개의 파티션으로 분리된다. RDD의 생성은 외부로부터 데이터를 로딩하거나 또는 코드에서 생성된 데이터를 저장함으로써 생성 할 수 있다.

RDD에서는 두 가지 오퍼레이션 Transformation과 Action을 지원한다.

• Transformation : 기존의 RDD데이터를 변경하여 새로운 RDD 데이터를 생성한다. filter와 같은 특정 데이터만 뽑아내거나 map 함수처럼, 데이터를 분산 배치 등을 한다.

• Action : RDD값을 기반으로 무엇인가를 계산(computation)하여 결과를 생성하는 것으로 예로는 count()와 같은 operation들을 들 수 있다.

그림 1은 RDD 모델을 나타내는 그림이다.

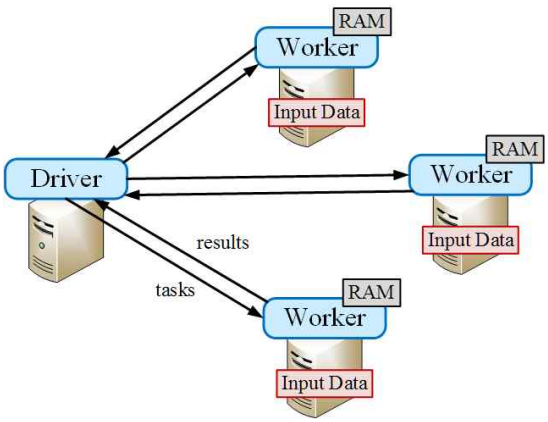

Fig. 1. Model of RDDs

III. The Proposed Scheme

본 논문에서는 하나의 마스터 노드와 두 개의 데이터 노드 총 3개의 노드가 있는 클러스터를 사용했다. 또한 input 데이터로 1024MB Wikipedia dump를 이용하여 WordCount를 스파크 버전 1.5.1에서 실행하여 실험을 실시했다.

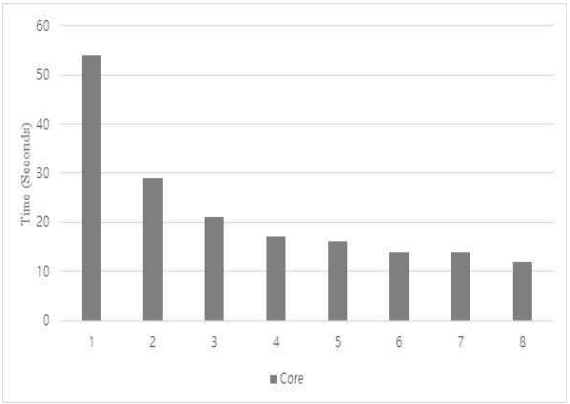

Fig. 2. Time Prediction for WordCount of CPU core count

그림 2는 core 1에서 core 8까지의 CPU core 수에 따른 WordCount 실행 시간을 나타낸다. CPU core 수에 따른 시뮬레이션 결과 core 1에서는 54초, core 2에서는 29초, core 3에서는 21초, core 4에서는 17초, core 5에서는 16초, core 6에서는 14초, core 7에서는 14초, core 8에서는 12초가 나왔다.

IV. Conclusions

본 논문에서는 아파치 스파크 플랫폼의 특정 작업 성능이 입력 데이터의 유형과 크기, 디자인 및 알고리즘 구현 및 컴퓨팅 능력에 따라 메모리 사용량 및 I/O 비용이 크게 달라질 수 있다는 문제점을 해결하기 위해 스파크 플랫폼에 대한 높은 정밀도 작업 성능을 예측할 수 있도록 CPU core 수의 증가에 따른 WordCount 시뮬레이션을 비교 평가 하였다. 비교 평가 결과 CPU core 2의 WordCount 실행 시간이 CPU core 1에서의 시간보다 반이상 줄어들고 core 수가 늘어날수록 시간이 감소하지만 core 5부터 core 8까지 줄어드는 시간은 core 1에서 core 2로 증가했을 때와 비교 했을 때 에는 큰 차이가 없었다.

Acknowledgment

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

References

[1] M. Zaharia, M. Chowdhury, M. J. Franklin, S. Shenker, and I. Stoica, “Spark: cluster computing with working sets,” in Proceedings of the 2nd USENIX conference on Hot topics in cloud computing, 2010.

[2] M. Zaharia, M. Chowdhury, T. Das, A. Dave, J. Ma, M. McCauley, M. J. Franklin, S. Shenker, and I. Stoica, “Resilient distributed datasets: A fault-tolerant abstraction for in-memory cluster computing,” in Proceedings of the 9th USENIX conference on Networked Systems Design and Implementation. USENIX Association, 2012, pp. 2–2.

[3] P. Patel, D. Bansal, L. Yuan, A. Murthy, A. Greenberg, D. A. Maltz, R. Kern, H. Kumar, M. Zikos, H. Wu et al., “Ananta: Cloud scale load balancing,” in ACM SIGCOMM Computer Communication Review, vol. 43, no. 4. ACM, 2013, pp. 207–218.

[4] S. Chaisiri, B.-S. Lee, and D. Niyato, “Optimization of resource provisioning cost in cloud computing,” Services Computing, IEEE Transactions on, vol. 5, no. 2, pp. 164–177, 2012.