

Hadoop 상에서 MapReduce 응용프로그램 평가

(Performance Evaluation of MapReduce Application running on Hadoop)

김준수[‡]
(Junsu Kim)

강윤희[§]
(Yunhee Kang)

박용범[¶]
(Youngbom Park)

요약 다양한 분야에서 빠르게 대용량의 자료가 생성됨에 따라 이를 처리하기 위해 분산 프로그래밍 모델인 MapReduce의 활용이 도입되고 있다. 본 논문에서는 SUN Blade150에 Solaris와 Linux 환경의 클러스터 시스템을 구축한 뒤 해당 환경에서의 MapReduce 미들웨어인 Hadoop 에서 응용수행에 대한 평균 시간 및 표준 편차를 평가하여 Hadoop 기반 MapReduce 구현이 어떠한 클러스터 시스템에 의해 성능이 영향을 미치는지를 보인다.

키워드 MapReduce, 성능 평가, 클러스터 시스템, Hadoop

Abstract According to the growth of data being generated in man fields, a distributed programming model MapReduce has been introduced to handle it. In this paper, we build two cluster system with Solaris and Linux environment on SUN Blade150 respectively and then to evaluate the performance of a MapReduce application running on MapReduce middleware Hadoop in terms of its average elapse time and standard deviation. As a result of this experiment, we show that the overall performance of the MapReduce application based on Hadoop is affected by the configuration of the cluster system.

Key words MapReduce, Performance evaluation, Cluster system, Hadoop

1. 서론

다양한 분야에서 빠르게 대용량의 자료가 생성되고 대용량에 대한 데이터 처리 요구가 증가하고 있으나 현재 디스크에 대한 접근은 대용량의 경우 데이터 전송 시 병목 문제를 겪는다[4,6]. 이러한 문제점을 개선하기 위해 분산 컴퓨팅을 지원하기 위한 목적으로 개발된 소프트웨어 프레임워크인 MapReduce의 활용이 증가하고 있다[2].

MapReduce는 다양한 분산 컴퓨팅 환경에서 동작하기 때문에 노드들 사이의 네트워크 환경, 운영체제의 차이 등 환경적인 요소에 의해 실행

속도 차이가 발생한다. 본 논문에서는 플랫폼 및 운영체제에 따른 분산 컴퓨팅 성능 분석을 위해 Solaris와 Linux 환경에서 성능 측정 및 분석을 수행하며, 실행 환경으로는 Solaris와 Linux 환경에서 MapReduce 응용인 WordCount의 성능을 비교하고자 한다. Solaris와 Linux는 같은 Unix 시스템을 기반으로 만들어진 운영체제이나 MapReduce 응용의 성능 차이가 존재 할 수 있으므로, MapReduce 응용 어플리케이션 중 하나인 WordCount를 사용하여 실험을 진행하며, Combiner의 사용 유무로 인한 두 운영체제의 속도와 안정성에 대해 비교 및 분석을 한다. 실험 환경으로는 Sparc 프로세서를 사용하며, Solaris 버전 10, Linux 배포판으로 Debian 버전 6.0.4를 사용하여 진행한다. 실험에 사용하는 MapReduce는 Apache의 오픈소스 프로젝트인 Apache Hadoop[7]을 이용하여 구성 하고자 한다.

[‡] 비 회 원 : 단국대학교 컴퓨터학과, tomy2174@gmail.com

[§] 비 회 원 : 백석대학교 정보통신학부 조교수, yunh.kang@gmail.com

[¶] 회 원 : 단국대학교 컴퓨터학과 교수, ybpark@dankook.ac.kr

논문의 구성은 2장에서 본 논문의 기초가 되는 MapReduce와 Combiner에 대한 관련 연구에 대해 기술하고, 3장에서는 Debian과 Solaris 클러스터 간의 MapReduce 응용에 대한 수행 시간 및 표준 편차에 대한 결과를 분석한다. 마지막 4장에서는 운영체제 환경에 따른 성능 분석 결과를 논하고 Hadoop 시스템 구축에 필요한 환경적 요소를 정의한 후 향후 연구 방향에 대하여 기술한다.

2. 기본 개념

2.1. MapReduce

MapReduce는 Google에서 제안한 분산 컴퓨팅 지원 목적의 프레임워크로서 주어진 문제를 추상화 해서 키와 값에 대한 계산으로 변환한 프로그래밍 모델이다[1,5]. MapReduce는 Map과 Reduce 단계로 구성된다. Map에서는 입력 데이터에서 키와 값을 추출하여 이 추출된 데이터를 Reduce에서 키 기준으로 값을 연산하여 분산 파일 시스템에 결과를 출력한다. Map 함수와 Reduce 함수 수행을 위한 Mapper 태스크와 Reducer 태스크는 병렬로 데이터를 처리하며, 별도로 시작하여 상호간의 의존성은 없다. MapReduce는 추상화 되어 있기에 프로그래머는 키와 값에 대한 고려만 하여 이를 작성할 수 있으며, 데이터 흐름은 내부 플랫폼에서 담당한다. 그림 1은 MapReduce의 내부 흐름도를 보인 것이다.

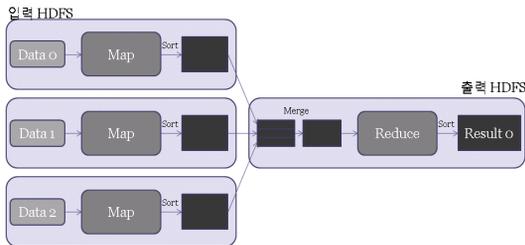


그림 1 MapReduce 내부 흐름도

2.2 Combiner

Combiner는 미니 리듀서라고도 불리며 MapReduce 응용을 수행 하는 과정에서 Map 작업에 필요한 데이터들이 어느 한 작업노드에 편중이 되어 있으면 Mapper 태스크에서 Reducer 태스크로 키와 값들을 전달할 경우 트래픽 폭주로 인하여 MapReduce 응용의 수행 속도가 느려지거나 네트워크가 stall 되는 상황이 일어날 수 있다. 이러한 문제를 해결 하기 위하여 Map 작업과 Reduce 작업 사이에 Combiner라는 단계 있으며 Reduce 작업을 도와 준다[3]. 그림 2는 Hadoop 미들웨어에서 Combiner의 위치를 기술한 것이다.

Combiner는 같은 노드의 Mapper 태스크에서 키와 값들을 받아 현재 작업 노드에서 자체적으로 Reduce 작업을 실행을 한 뒤 Reducer 태스크에게 Combine 작업을 한 결과인 키와 값을 전달한다. Combiner를 사용하게 될 경우 네트워크 트래픽이 크게 감소될 수도 있다. 그러나 값들의 연산이 교환법칙 혹은 결합법칙이 성립되지 않을 경우에는 값이 변경이 될 수 있으므로 사용하지 않는 것이 적절하다.

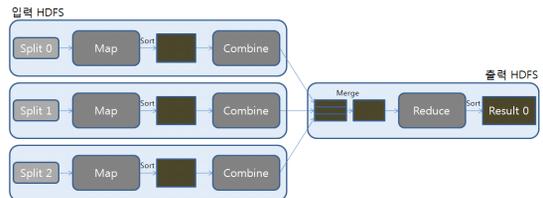


그림 2 Combiner를 사용한 MapReduce 내부 흐름도

3. 성능평가

3.1. 실험환경 구축

작성된 Hadoop 응용은 다양한 범용 컴퓨팅 자원에서 작업을 수행한다. 다음에 나오는 표 1은 실험 환경의 하드웨어 플랫폼 및 운영체제 구성을 기술한 것이다.

표 1. 실험을 위한 컴퓨팅 구성

	Type-1	Type-2
CPU 사양	Sparc Processor 650Mhz * 1 ~ 12	
코어 수	1 ~ 12	
메모리크기	512Mbyte * 1 ~ 12	
운영체제	Solaris 버전 10	Debian 버전 6.0.4
MapReduce 미들웨어	Hadoop 1.0.2	

이 실험에서는 작업 노드의 구성에 따라 Type-1과 Type-2로 구분하여 사용하였다. Type-1은 Solaris 10 버전을 사용하며, Type-2에는 Debian 6.0.4 를 사용하며, 하드웨어는 같은 기종의 서버들을 사용한다. 또한 Type-1과 Type-2의 마스터 노드는 별도의 서버에서 수행하도록 분산한다.

3.2 결과 및 성능평가

성능평가를 위하여 507MB 크기의 임의의 텍스트 파일을 이용하여 작업 노드가 1대인 경우부터 12대인 경우까지 각각 10회씩 WordCount 수행하여 수행 시간의 평균값 및 표준편차에 대하여 평가한다. <그림 3>은 노드가 추가됨에 따른 MapReduce 의 응용 수행의 응답 시간을 보인 것이다.

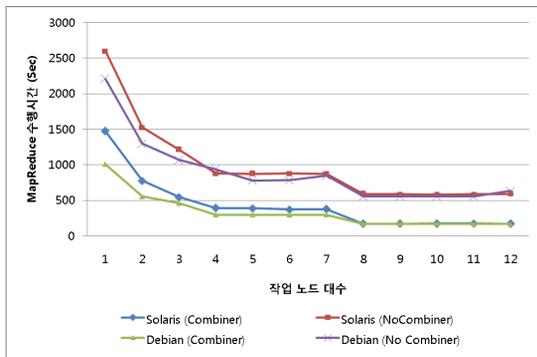


그림 3 Debian과 Solaris 환경에서의 MapReduce 응용 수행 시간

표 2는 MapReduce 의 응용 시간을 보인 것으로 실험을 통해 Combiner를 사용할 경우 Combiner를 사용 하지 않은 경우에 비하여 MapReduce 응용 수행 시간이 Solaris는 44.11%, Debian은 37.71% 더 적어진 것을 볼 수 있다. 그러나 Combiner를 사용하지 않은 경우에는 작업 노드로 4대를 사용한 경우와 12대를 사용한 경우에는 MapReduce 응용의 평균 수행 시간이 Debian 보다 Solaris가 더 적어진 것으로 나왔으며, 7대를 사용한 경우에도 작업 노드를 6대 사용한 경우보다 상대적으로 MapReduce 응용 수행 시간이 상대적으로 증가 하였다. 이러한 이유는 각각의 작업 노드를 4대, 7대, 12대로 증가하여 실험을 하는 경우 MapReduce 작업이 실패하여 시간이 지연이 된 것이다.

표 2. Debian과 Solaris 환경에서의 MapReduce 응용 수행 시간

(단위: 초)

작업노드 대수	Solaris (Combiner)	Solaris (No Combiner)	Debian (Combiner)	Debian (No Combiner)
1	1474.5	2599.9	1007.6	2207.3
2	777.1	1524.0	554.5	1297.4
3	547.9	1213.6	458.8	1067.4
4	393.0	879.5	297.4	936.9
5	387.7	874.1	298.0	779.5
6	374.5	879.2	293.9	783.5
7	374.7	875.8	296.5	847.5
8	171.1	588.3	171.0	552.9
9	172.8	586.3	171.9	554.0
10	173.6	582.5	172.8	553.1
11	174.8	584.8	168.3	553.0
12	172.8	587.8	169.0	634.3
합계	5194.5	11775.8	4059.7	10766.8
평균	432.9	981.3	338.3	897.2

그림 4로부터 Debian은 Combiner를 사용 하지 않은 경우는 Combiner를 사용하였을 경우보다 표준 편차의 크기가 959.28%로 증가한 것을 볼

수 있었다. 그러나 Solaris에서 Combiner를 사용하지 않은 경우와 Combiner를 사용한 경우의 표준편차를 비교하면 83.94%로 오히려 감소한 것을 볼 수 있다.

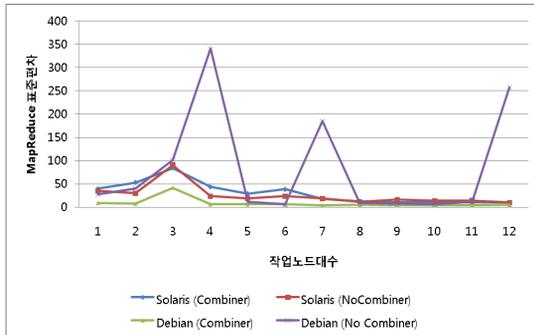


그림 4 MapReduce 응용 수행 시간에 대한 표준편차

표 3은 MapReduce 응용 시간의 표준편차를 보인 것으로 Debian 환경에서 Combiner를 사용하는 경우가 응용수행 시간의 편차가 적은 것을 보이고 있으며 Combiner를 사용하지했을 때 가장 높은 편차 값을 보인다.

표 3. MapReduce 응용 수행 시간에 대한 표준편차

(단위: 초)

작업노드 대수	Solaris (Combiner)	Solaris (No Combiner)	Debian (Combiner)	Debian (No Combiner)
1	39.4863	34.7321	9.2280	26.6252
2	53.1736	29.4165	7.4722	39.9811
3	83.9093	90.1359	41.8484	100.8444
4	43.7061	23.2869	6.6366	340.4365
5	29.1892	18.9059	6.1464	10.8858
6	38.2281	23.5504	5.8013	5.8926
7	18.0373	18.2257	4.1700	184.0551
8	13.0678	11.1360	4.8762	8.1302
9	10.3688	16.3371	4.2804	6.6667
10	11.3744	13.1255	4.3665	6.5056
11	10.2610	14.0539	3.8601	10.6249
12	10.0532	9.9867	5.2915	256.7857
합계	360.9	302.9	104.0	997.4
평균	30.1	25.2	8.7	83.1

4. 결론

우리는 앞서 Solaris 와 Linux 간의 Combiner를 사용 하였을 경우와 사용하지 않은 경우에 대하여 MapReduce 응용에 대한 수행 속도와 안정성의 차이를 살펴보았다. 이 실험에서 Combiner를 사용하고 Solaris와 Linux의 MapReduce응용 수행 시간을 측정 한 결과 Linux가 Solaris 보다 상대적으로 안정적이고 전체적으로 적은 시간이 소요 된 것을 볼 수 있었다.

Combiner를 적용 시키지 않고 실험한 결과 또한 전반적으로 Linux가 Solaris에 비해 시간이 적게 소요가 되었으나 안정적인 측면에서 Solaris는 MapReduce 응용 수행에 실패한 경우가 없으나 Linux는 작업 노드의 수가 4대, 7대, 12대인 경우 MapReduce 응용 수행이 실패한 결과가 전체 작업 시간에 영향을 미쳐 Linux는 Solaris 보다 수행 시간이 큰 것을 관찰 하였다.

본 논문에서는 MapReduce를 구동할 경우 Linux를 사용하면 속도가 Solaris보다 높은 속도를 낼 수 있으나 Combiner를 사용하지 않아 네트워크 리소스를 많이 사용하게 될 경우에는 Linux보다 Solaris가 더 안정적인 것을 볼 수 있었다.

본 논문에서는 MapReduce 측정 실험으로 적은 용량의 파일로 WordCount 응용을 사용하여 작업을 수행 하였으나 더 대용량 파일 및 다른 MapReduce 응용에서의 성능평가를 수행할 예정이다.

참 고 문 헌

[1] J. Dean and S. Ghemawat, "Mapreduce: Simplified Data Processing on Large Clusters," Communications of the ACM, Vol. 51(1), pp. 107-113, Jan., 2008.

- [2] 강윤희, “FutureGrid 클라우드 컴퓨팅 환경에서의 Twister 수행 MapReduce 응용 구성“, 한국 정보 기술 학회 논문지 제9권 제4호, 147-154, 2011.4
- [3] Chuck Lam, “Hadoop in Action“, Manning, 2010
- [4] Tom White, “Hadoop: The Definitive Guide, 2nd Edition“, O’Reilly Media / Yahoo Press, 2010
- [5] J. Dean and S. Ghemawat, MapReduce: A Flexible Data Processing Tool. CACM, vol. 53, pp. 72-77, 2010
- [6] K. Morton, A. Friesen, M. Balazinska and D. Grossman, Estimating the Progress of MapReduce Pipelines, IEEE 26th International Conference on Data Engineering (ICDE), 2010, pp. 681 - 684, Long Beach, CA, 2010
- [7] Hadoop, <http://hadoop.apache.org>



강 윤희

1989년 동국대학교 컴퓨터공학과(공학사)
 1991년 동국대학교 컴퓨터공학과(공학석사)
 2002년 고려대학교 컴퓨터과학과(이학박사)
 2000년~현재 백석대학교 정보통신학부 조교수

<관심분야> 클라우드 컴퓨팅, 그리드 컴퓨팅, 결합포용



박 용 범

1991년 N.Y. Polytechnic University Computer Sci. & Eng. Ph.D.

1993년~현재 단국대학교 컴퓨터과학 교수

<관심분야> 지능형 소프트웨어 공학, 보안 소프트웨어 개발

저 자 소 개



김 준 수

2011년~현재 단국대학교 컴퓨터과학과 재학

<관심분야> 클라우드 컴퓨팅, 분산 처리 시스템