맵리듀스 프레임워크의 중간 데이터가 성능에 미치는 영향에 관한 연구

김신규, 엄현상, 염헌영
서울대학교 컴퓨터공학부
e-mail:{sgkim, hseom, yeom}@dcslab.snu.ac.kr

A Study on the Effects of Intermediate Data on the Performance of the MapReduce Framework

Shin-gyu Kim, Hyeonsang Eom, Heon Y. Yeom
School of Computer Science and Engineering, Seoul National University

요 약

맵리듀스 프레임워크는 개발의 편의성, 높은 확장성, 결함 내성 기능을 제공하며 다양한 대용량 데이터 처리에 사용되고 있다. 또한, 최근의 데이터의 폭발적 증가는 높은 확장성을 제공하는 맵리듀스 프레임워크의 도입의 필요성을 더욱 증가시키고 있다. 이 경우 하나의 단일 클러스터에서 처리할 수 있는 계산 용량을 넘어설 수 있으며, 이를 위하여 클라우드 컴퓨팅 서비스 등에서 계산자원을 빌려오게 된다. 하지만 현재의 맵리듀스 프레임워크는 단일 클러스터 환경을 가정하고 설계되었기에 여러 개의 클러스터로 이루어진 환경에서 수행시킬 경우 전체 계산자원의 이용률이 떨어져서 투입된 자원에 비해 전체적인 성능이 낮아지는 경우가 발생하게 된다. 본 연구에서는 이의 원인이 맵과 리듀스 단계 사이에 존재하는 중간결과의 전송에 있음을 밝히고, 이의 전체 맵리듀스 프레임웍의 성능에 미치는 영향에 대하여 분석해보았다.

1. 서론

최근 IT 분야 최고의 화두 중 하나로 빅 데이터 (Big Data) 의 효율적인 처리방법을 꼽을 수 있다. 이를 위하여 다양한 연구 및 개발이 진행되고 있으며, 그 중 구글 (Google) 에서 개발한 맵리듀스 (MapReduce) 프레임워크가 많은 관심을 받고 있다. [1] 맵리듀스가 주목을 받고 있는 이유는 기존의 분산병렬처리 프레임워크에서 부족했던 점인 개발의 편의성, 높은 확장성, 결함 내성 기능을 모두 제공하기 때문이다. 이에 아파치 (Apache) 공개 소프트웨어 재단은 야후! (Yahoo!) 의 지원을 받아 맵리듀스의 공개버전인 하둡 (Hadoop) [2] 을 개발하여 배포하고 있으며, 현재 IT서비스뿐만 아니라 과학계산 등의 다양한 분야에서 사용되고 있다. [4, 5]

맵리듀스 프레임워크는 높은 확장성을 제공하기에 클러스터에 계산자원을 추가하여 쉽게 처리능력을 확장할 수 있다. 하지만, 클러스터에 계산자원을 추가하는 일은 장기적인 계획 하에 이루어져야하기에 예측하지 못한 급격한 작업량의 증가에 대처하기에는 어려움이 있다. 이에 대한 가장 현실적인 대응책으로는 클라우드 컴퓨팅 서비스 (Cloud Computing Service) 를 꼽을 수 있으며, 이로부터 부족한 계산자원을 빌려와서 원하는 시기에만 처리능력을 확장할 수 있게 된다. 클라우드 컴퓨팅 서비스를 이용하여 처리능력을 확장시킬 경우 클러스터에 계산 자원을 구입하여 추가하는 것보다 도입 시간을 줄일 수 있을 뿐만 아니라 비용적인 측면에서도 이득을 가져다 줄 수 있다.

대표적인 클라우드 컴퓨팅 서비스인 아마존 (Amazon) 의 EC2 (Elastic Compute Cloud) [3] 는 이미 하둡을 이용하여 맵리듀스를 사용할 수 있는 서비스를 이미 제공하고 있다. 하지만 이는 아마존 EC2만으로 구성된 맵리듀스 프레임워크만을 지원하기에 기관이 보유중인 클러스터의 계산 용량을 늘리는 용도로는 사용이 어렵다. 이 경우에는 원하는 만큼의 가상머신을 할당받아서 계산자원을 추가해야 한다. 이 경우 사설 클러스터와 클라우드는 속도가 느린 WAN (Wide Area Network) 로 연결되게 된다. 맵리듀스 프레임워크는 애초에 설계될 당시 단일 클러스터 환경을 가정하고 개발되었기 때문에 계산 노드의 성능에 차이가 있거나 노드 간 네트워크 성능에 많은 차이가 존재할 경우 전체적인 성능이 예상한 만큼 나오지 않는 문제가 있음이 알려져 있다.

본 연구에서는 맵리듀스 프레임워크가 여러 클러스터를 사용하여 구성된 경우[1] 성능 하락의 원인이 맵 (Map) 과 리듀스 (Reduce) 단계의 사이에 존재하는 중간 데이터 (Intermediate Data) 의 전송 속도에 있음을 밝히고, 중간 데이터의 전송 속도가 전체적인 성능에 어떠한 영향을 미

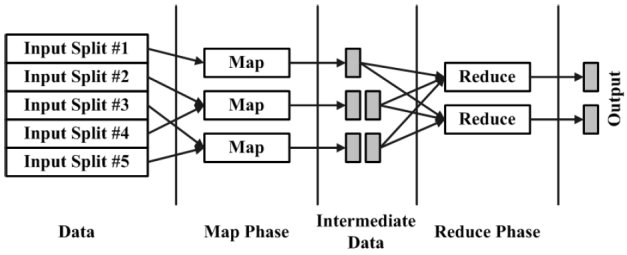
1) 여러 개의 클러스터를 이용하는 경우를 인터클라우드 (Inter-cloud) 환경이라고도 한다.

치는지 분석해본다.

2. 인터클라우드 환경에서의 맵리듀스

2.1. 맵리듀스 프레임워크의 구조

맵리듀스 프레임워크는 구조는 그림 1과 같다. 전체적으로 맵과 리듀스 단계로 이루어져 있으며, 사용자의 입력 데이터는 먼저 맵 단계에서 처리가 된다. 맵 단계에서는 사용자의 입력 데이터를 받아서 키 (Key)와 값 (Value)로 이루어진 쌍 (Pair)를 생성하게 되는데 이것이 바로 중간 데이터이며 다음 단계인 리듀스 단계로 넘어가게 된다. 리듀스 단계에서는 키 별로 데이터가 처리되며 최종적인 결과를 생성해낸다.

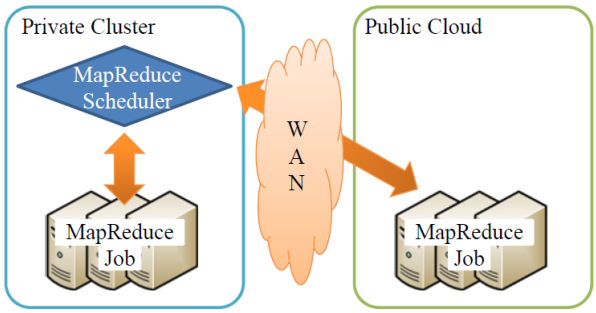


(그림 1) 맵리듀스 프레임워크의 구조

맵과 리듀스 단계에서는 각 프로세스가 독립적으로 실행되며, 이로 인해 맵리듀스 프레임워크는 높은 확장성을 갖게 된다. 하지만 맵 단계와 리듀스 단계 사이에는 보이지 않는 벽 (Barrier) 이 존재하는데 이는 리듀스 단계가 시작되기 위해서는 이전 단계인 맵 단계에서 모든 프로세스가 수행을 끝마쳐야 하기 때문이다. 이런 특성으로 인하여 맵리듀스는 추론적 실행 (Speculative Execution) 과 같은 기법을 이용하여 Barrier에 의한 성능 저하를 막고자 한다.

2.2. 인터클라우드 환경

인터클라우드 환경은 그림 2와 같이 사설 클러스터와 공개 클라우드 서비스로 구성된다. 둘 사이는 LAN (Local Area Network) 에 비해 낮은 성능을 갖는 WAN으로 연결되어 있다. 이 때 맵리듀스 스케줄러와 사용자 입력 데이터는 모두 사설 클러스터에 존재한다. 그 이유는



(그림 2) 인터클라우드 환경에서의 맵리듀스 프레임워크

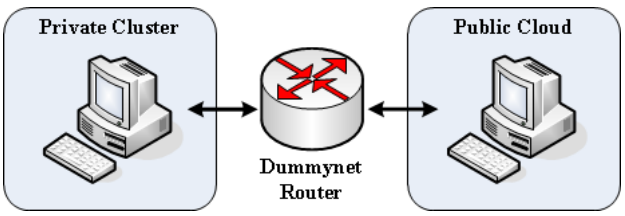
클라우드로 대용량을 데이터를 전송하는 데에는 많은 시간이 걸리는데다가, 맵리듀스는 전체 데이터가 없어도 바로 맵 단계의 처리를 시작할 수 있기 때문이다. 따라서 인터클라우드 환경에서 맵리듀스 프레임워크는 그림 2와 같이 구성될 수 있다.

인터클라우드 환경에서 구성된 맵리듀스 프레임워크의 처리 성능에서 가장 큰 병목은 사설 클러스터와 공개 클라우드 간의 WAN에서 발생한다. 앞서 설명했듯이 리듀스 단계는 맵 단계가 모두 끝나야 시작될 수 있기에 WAN을 거쳐서 전송되는 Intermediate Data의 전송 시간이 길어지면 전체 수행시간이 늘어날 수밖에 없다. 본 연구에서는 이의 영향에 대해서 분석해보고자 한다.

3. 실험 및 분석

3.1. 실험방법

실험은 그림 3과 같이 두 대의 머신을 이용하여 진행하였으며, 두 대의 머신은 FreeBSD 6.0 기반의 더미넷 라우터 (Dummynet Router) 를 통해서 연결되었다. 더미넷 라우터를 통해서 WAN 환경을 에뮬레이션 하였으며 대역폭은 25Mbps 부터 100Mbps 까지 변경시켰다. 첫 번째 머신은 인텔의 Core2-Quad 2.83Ghz CPU와 8GByte의 메모리를 탑재하고 있으며, 두 번째 머신은 인텔의 Xeon 2.0Ghz CPU 두 개와 16GB의 메모리를 탑재하고 있다. 운영체제는 모두 Ubuntu 8.04 (Linux Kernel 2.6.24) 이며, 모든 맵리듀스 프로세스는 Xen 가상화 환경에서 실행되었다. 첫 번째 머신을 사설 클러스터 환경이라고 가정하였고, 두 번째 머신을 공개 클라우드 서비스라고 가정하였다.



(그림 3) 실험 환경 구성

맵리듀스 프레임워크는 하둡 0.19를 이용하였으며, 설정은 기본적으로 제공하는 내용을 따랐다. 실험은 맵리듀스의 성능을 평가하는데 가장 많이 사용되는 WordCount 프로그램을 기반으로 Intermediate Data의 양을 조절할 수 있도록 수정하였다. 각 머신의 맵 함수를 수행하는 프로세스 (Mapper) 와 리듀스 함수를 수행하는 프로세스 (Reducer) 의 개수는 각 4개씩 설정하였고, 공개 클라우드 쪽에는 Reducer를 배치하지 않았다. 사용된 입력 데이터의 크기는 약 5Gbyte 였다.

3.2. 실험결과

본 절에서는 다양한 환경에서의 실험결과를 보인다. 먼저 WAN의 대역폭이 100Mbps 인 경우이다. 입력 데이터 대비 중간 데이터의 비율을 1%부터 100%까지 변경시키면서 사설 클러스터와 공개 클라우드의 처리속도를 측정하였으며, 각 경우에 대해서 클라우드에서 처리된 데이터의 양을 측정하였고, 그 결과를 아래의 표 1에 나타내었다.

<표 1> 100Mbps WAN으로 연결된 경우의 성능

중간 데이터 비율 (%)	처리 속도 (Mbps) Node 1	Node 2	클라우드에서 처리된 데이터량 (Mbyte)	중간 데이터의 크기 (Mbyte)
1	326.9	90.9	1196.7	23.0
10	221.8	87.3	1495.3	288.1
20	154.2	76.4	1708.7	658.4
30	109.9	63.7	1942.3	1130.9
40	86.0	53.7	2008.3	1537.3
50	69.3	45.2	2050.0	1985.3
60	58.8	38.7	2050.0	2369.3
70	49.8	33.6	2092.7	2844.0
80	43.6	30.7	2155.7	3334.3
90	38.0	26.3	2113.0	3697.3
100	34.5	25.0	2177.0	4183.7

위의 표 1에서 볼 수 있듯이 중간 데이터의 크기가 작은 경우에는 대부분의 데이터가 사설 클러스터에서 처리되지만, 중간 데이터의 비율이 높아짐에 따라 공개 클라우드에서 처리되는 데이터의 양이 점차 늘어남을 볼 수 있다. 그 이유는 중간 데이터가 늘어남에 따라 CPU에서 처리되어야할 작업의 양이 늘어나게 되고, 상대적으로 네트워크의 병목현상이 전체 성능에 미치는 영향이 줄어들기 때문이다. 여기서 주목해야할 구간은 중간 데이터의 비율이 30% 이하인 경우이다. 중간 데이터의 비율이 30%인 경우 사설 클러스터에서의 처리속도가 109.9Mbps로 측정이 되었는데, 이 경우 WAN의 대역폭보다 높은 처리속도를 보이므로, 공개 클라우드에서의 처리 속도는 WAN의 제한을 받게 된다. 하지만, 아래 결과에서 볼 수 있듯이 WAN의 대역폭 100Mbps보다 낮은 63.7Mbps를 처리능력을 보여주고 있다. 이는 공개 클라우드에서 처리된 맵 단계의 결과인 중간 데이터를 다시 사설 클라우드로 보내는 작업이 입력 데이터를 공개 클라우드로 보내는 작업과 겹치면서 발생하는 현상이다. 하지만 이런 현상은 중간 데이터의 비율이 높아짐에 따라 점차 완화되는 모습을 관찰할 수 있었다. 이는 계산량이 늘어남에 따라 입력 데이터의 처리 속도가 감소하면서 최대의 성능을 얻기 위하여 필요한 대역폭이 감소하여 중간 데이터를 보내는 트래픽과 간섭이 줄었기 때문이다.

위와 같은 내용의 실험을 WAN의 대역폭이 50Mbps와 25Mbps의 경우에도 수행하였으며 그 결과를 아래의 표 2와 표3에 나타내었다.

<표 2> 50Mbps WAN으로 연결된 경우의 성능

중간 데이터 비율 (%)	처리 속도 (Mbps) Node 1	Node 2	클라우드에서 처리된 데이터량 (Mbyte)	중간 데이터의 크기 (Mbyte)
1	330.6	47.1	705.0	13.6
10	224.0	46.7	963.0	185.5
20	158.3	44.5	1196.7	461.1
30	111.0	41.8	1475.0	858.2
40	88.3	38.1	1579.7	1209.3
50	70.7	34.7	1708.7	1654.7
60	59.2	31.7	1836.7	2122.7
70	50.3	28.2	1878.3	2553.0
80	43.7	25.8	1923.0	2974.0
90	38.7	23.7	1943.3	3399.3
100	35.0	22.1	2008.3	3859.0

<표 3> 25Mbps WAN으로 연결된 경우의 성능

중간 데이터 비율 (%)	처리 속도 (Mbps) Node 1	Node 2	클라우드에서 처리된 데이터량 (Mbyte)	중간 데이터의 크기 (Mbyte)
1	330.6	24.7	406.3	7.8
10	226.4	24.4	579.0	111.5
20	158.0	23.5	771.0	297.0
30	113.0	21.7	899.0	522.8
40	88.8	21.5	1048.3	802.3
50	71.6	20.4	1196.7	1158.7
60	61.9	19.4	1283.0	1482.0
70	53.4	18.2	1366.3	1857.0
80	46.6	17.8	1475.0	2281.0
90	41.9	17.7	1579.7	2764.0
100	37.7	16.9	1643.7	3158.7

이 경우에도 역시 100Mbps의 경우와 같은 결과를 경향을 보여주고 있음을 확인할 수 있다. 위의 실험들을 통해서 맵리듀스를 인터클라우드 환경에서 수행할 경우 중간 데이터의 전송이 맵 단계의 입력 데이터의 전송을 방해하여 공개 클라우드에서 실행되는 맵 프로세스의 성능을 저해하는 영향이 있음을 알아내었고, 인터클라우드 환경에서 가장 큰 이득을 얻을 수 있는 맵리듀스 프로그램은 데이터 집약적인 (Data-intensive) 프로그램보다는 계

산 집약적인 (Computation-intensive) 프로그램이 훨씬 높은 이득을 달성할 수 있음을 밝혀내었다.

4. 결론

본 연구에서는 여러 개의 클러스터로 구성된 인터클라우드 환경에서 맵리듀스 프레임워크를 구축할 때 낮은 성능의 WAN이 전체적인 성능에 어떠한 영향을 미치는지를 다양한 환경에서 수행된 실험들을 통하여 밝혀내었다. 클라우드 서비스는 사용한 시간에 따라 비용을 지불하는 서비스이기 때문에 비용대비 효과를 극대화하는 것이 중요하며, 이를 위해서는 데이터 집약적인 작업보다는 계산 집약적인 작업을 인터클라우드 환경에서 수행하는 것이 바람직하다고 할 수 있다.

감사의 글

이 논문은 2011년도 정부(교육과학기술부)의 재원으로 한국연구재단-차세대정보컴퓨팅기술개발사업의 지원을 받아 수행된 연구임. (No. 2011-0020521) 이 연구를 위해 연구장비를 지원하고 공간을 제공한 서울대학교 컴퓨터연구소에 감사드립니다.

참고문헌

[1] J. Dean, and S. Ghemawat, “MapReduce: Simplified Data Processing on Large Clusters,” in Proceedings of Sixth Symposium on Operating System Design and Im-plementation, 2004.

[2] Apache, Hadoop. http://hadoop.apache.org/.

[3] Amazon, EC2 service. http://aws.amazon.com/ec2.

[4] C.T. Chu, S.K. Kim, Y.A. Lin, Y. Yu, G.R. Bradski, A.Y. Ng, K. Olukotun. Map-Reduce for Machine Learning on Multicore, in Proceedings of Neural Information Processing Sys-tems Conference (NIPS) (2006)

[5] G. Fox, X. Qiu, S. Beason, J. Choi, J. Ekanayake, T. Gunarathne, M. Rho, H. Tang, N. Devadasan, G. Liu. Biomedical case studies in data intensive computing, in Proceedings of the 1st International Conference on Cloud Computing (2009)