한국컴퓨터정보학회 동계학술대회 논문집 제23권 제1호 (2015. 1)

하둡 맵리듀스 성능 향상을 위한 데이터 프리패칭과 스트리밍

이정준°, 김경태*, 윤희용*
°*성균관대학교 정보 통신대학
e-mail: jungjune86@skku.edu°, kyungtaekim76@gmail.com*, youn7147@skku.edu*

Data Prefetching and Streaming for Improving the Performance of Mapreduce of Hadoop

Jung June Lee°, Kyung Tae Kim*, Hee Yong Youn*
°*College of Information and Communication Engineering, Sungkyunkwan University

● 요 약 ●

최근 소셜 네트워크, 바이오 컴퓨팅, 사물 인터넷 등의 출현으로 인해 기존의 IT환경보다 많은 데이터가 생성되고 있고, 이로 인해 효율적인 대용량 데이터 처리기법에 대한 연구가 진행 되고 있다. 맵리듀스는 데이터 집약적인 연산 어플리케이션에 효과적인 프로그래밍 모델로써, 대표적인 맵리듀스 어플리케이션으로는 아파치 소프트웨어 재단에서 개발 지원중인 하둡 이 있다. 본 논문은 하둡 맵리듀스의 성능 향상을 위해 데이터 프리패칭 기법과 스트리밍 기법을 제안한다. 하둡 맵리듀스의 성능 이슈 중 하나는 맵리듀스 과정에서 입력 데이터 전송에 의한 작업 지연이다. 이러한 데이터 전송 시간을 최소화하기 위해, 기존 맵리듀스와는 달리 데이터 전송을 담당하는 프리패칭 스레드를 별도로 생성하였다. 그 결과 데이터의 맵리듀스 작업 중에도 데이터 전송이 가능하게 되어 전체 데이터 처리 시간을 줄일 수 있었다. 이러한 프리패칭 기법을 사용해도 하둡 맵리듀스의 특성상 최초 데이터 전송 시에는 작업대기를 하게 되는데, 이 대기시간을 줄이고자 스트리밍 기법을 사용하여 데이터 전송에 의한 대기시간을 추가로 줄일 수 있었다. 제안하는 기법의 성능을 측정하기 위해 수학적인 모델링을 하였으며, 성능 측정결과 기존의 하둡 맵리듀스 및 프리패칭 기법만 적용된 맵리듀스 보다 스트리밍 기법이 추가 적용된 맵리듀스의 성능이 향상되었음을 확인 할 수 있었다.

키워드: 맵리듀스(Mapreduce), 프리패칭(Prefetching), 스트리밍(Streaming)

I. 서 론

최근 소셜 네트워크, 바이오 컴퓨팅, 사물 인터넷 등의 출현으로 인해 기존의 IT환경보다 많은 데이터 가 생성되고 있고, 이로 인해 효율적인 대용량 데이터 처리기법에 대한 연구가 진행 되고 있다. 맵리듀스는 데이터 집약적인 연산 어플리케이션에 효과적인 프로그래밍 모델로써, 대표적인 맵리듀스 어플리케이션으로 아파치 소프트웨어 재단에서 개발 지원중인 하둡 이 있다 [1-3]. 하둡 맵리듀스 특성상 클러스터를 구성하는 모든 노드의 연산 자원을 최대한 활용하기 위해 로컬 데이터가 아닌 외부데이터를 처리 하는 경우, 맵리듀스의 효율을 위해 데이터 전송을 고려해야 한다. 이러한 데이터 전송 시간동안의 데이터 처리 지연을 해결하고자 데이터 프리패치 기법[5] 의 연구가 진행되었다. 하지만, 프리패칭 기법을 적용하여도 최초 데이터 전송시간 동안은 데이터 처리가 불가능하므로, 작업대기 시간이 존재한다.

본 논문에서는 데이터 처리와 전송을 동시에 가능하게 해주는 프리패칭 기법과, 데이터 전송 시간을 단축시킬 수 있는 스트리밍 기법을 제안함으로써, 하둡 맵리듀스의 작업 처리 성능을 향상시키고자 한다.

II. 관련 연구

1. Mapreduce

하둡 맵리듀스는 맵과 리듀스 두 단계로 구성된다. 그림 1 에 나타난 바와 같이, 맵 단계에서 복수개의 맵 태스크가 입력된 데이터를 스플릿으로 나누어 처리하고, 모든 맵 태스크가 종료되면 맵 단계는 종료된다. 이후 맵 태스크의 결과들이 섞이고, 정렬 된 후에 리듀스 태스크들에 의해 병렬로 처리되어, 리듀스 작업의 결과가 하나의 아웃풋 파일로 합쳐지면서 맵리듀스 전체 작업이 끝나게 된다[4]. 이 과정에서 맵 태스크를 처리하는 노드가 자신이 실행하는 맵 태스크의 입력 데이터 스플릿을 가지고 있지 않을 경우, 해당 입력파일을 가진 노드의 데이터를 전송받아 맵 태스크를 실행하게 되는데, 이러한 전송 중에는 처리 할 수 있는 데이터가 없으므로, 데이터 처리는

데이터 전송이 완료되기 전 까지 대기한다.

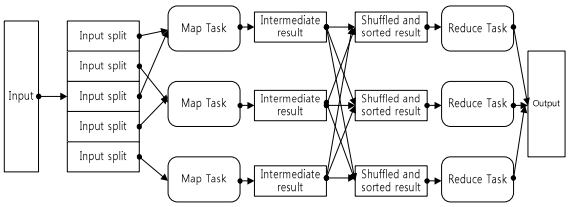

그림 1. 맵리듀스 작업 처리 흐름도
Fig. 1. Mapreduce Workflow

2. Data Prefetching

하둡 맵리듀스의 맵 태스크는 할당받은 맵 프로세스를 처리하기 위한 입력 데이터가 필요하며, 입력데이터가 로컬 노드가 아닌 다른 노드에 있을 경우, 데이터 전송을 통해 입력 데이터를 확보한다. 이러한 전송 과정 중에는 데이터를 처리 할 수 없으므로 전체 처리시간이 지연 되며, 이를 해소하고자 그림 2 와 같은 형태의 프리패치 기법이 제안되었다. [5]

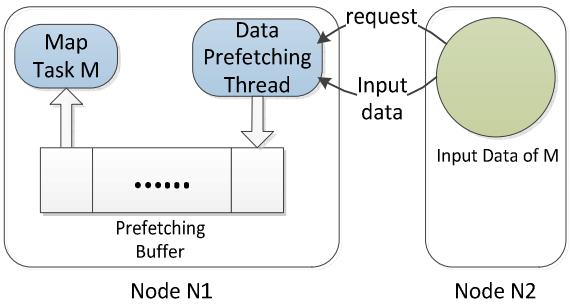

그림 2. 프리패치가 적용된 맵 태스크
Fig. 2. Map Task with Prefetch Architecture

프리패칭 기법의 핵심은 직렬로 처리되는 데이터 전송과 처리를 프리패칭 스레드를 이용하여 병렬로 처리함으로써 데이터 전송과 처리를 동시에 가능케 하여 전체 맵 태스크 처리 시간을 줄이는데 있다. 하지만 이러한 프리패칭 기법을 적용하더라도, 최초 데이터 전송 시간동안은 한 개의 데이터가 완전히 전송될 때 까지 데이터 처리를 실행 할 수 없으므로 작업대기 시간이 발생한다.

III. 본 론

본 논문에서는 하둡 맵리듀스의 성능 향상을 위해 프리패치 기법을 기반으로 데이터 전송 및 처리를 병렬화 하고, 추가로 최초 데이터 전송 중에 발생하는 작업대기 시간을 최소화하기 위해 스트리밍 기법[6]을 적용한 새로운 시스템을 제안한다. 기존의 프리패칭 기법은 프리패칭 스레드를 이용해 그림 3과 같이 데이터 처리와 전송을 동시에 수행 하지만, 최초의 데이터 전송 중 에는 데이터 처리를 할 수 없으므로 전송이 완료될 때 까지 작업대기시간이 존재한다.

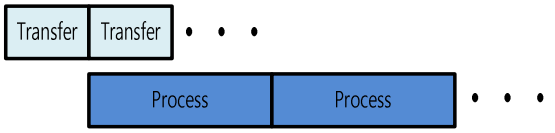

그림 3. 프리패칭이 적용된 맵태스크 작업 실행
Fig. 3. Maptask with Prefetch

하둡 맵리듀스 작업의 입력 데이터는 라인 단위로 나뉘어 맵 태스크에서 실행됨으로써[4] 입력파일 전체가 아닌 부분적인 데이터만으로도 맵 태스크의 수행이 가능하다. 이는 데이터 전송 중에 처리를 하는 스트리밍 기법과 적용이 가능함을 의미하며, 기존의 프리패칭 기법에 스트리밍 기법을 추가로 적용할 경우 맵 태스크는 그림 4와 같이 전체 작업 처리시간을 단축시킬 수 있다.

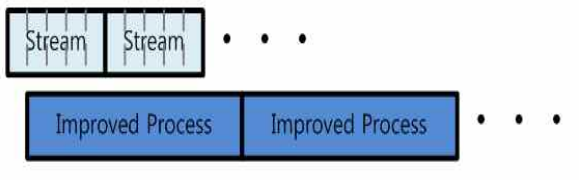

그림 4. 프리패치와 스트리밍이 적용된 맵태스크 작업 실행
Fig. 4. Maptask with Prefetch and Streaming

제안된 시스템과 기존 하둡의 성능 측정을 위한 수학적인 모델링을 위해, 아래의 표 1 과 같이 인자 값들을 정의 하였다.

표 1. 모델링 인자 값
Table 1. Modeling Factors

변수 명	설명
S_{map}	맵 태스크의 입력 스플릿 사이즈
S_{buf}	입력 스플릿이 저장되는 버퍼 사이즈
$S_{streambuf}$	입력 스플릿이 스트림으로 전송될 때 저장되는 버퍼 사이즈
V_{map}	기존 하둡에서 맵 태스크의 초당 데이터 처리 량
V'_{map}	향상된 하둡에서 맵 태스크의 초당 데이터 처리 량
V_{tran}	입력 데이터가 있는 노드와 맵 태스크가 진행되는 노드 사이의 초당 데이터 전송 속도
t_{hadoop}	기존 하둡이 맵 태스크를 처리하는데 걸리는 시간
$t_{prefetch}$	프리패칭 기법이 적용된 맵리듀스의 맵 태스크를 처리하는데 걸리는 시간
$t_{improved}$	프리패칭 기법과 스트리밍 기법이 적용된 맵리듀스의 맵 태스크를 처리하는데 걸리는 시간

위와 같은 인자 값을 통해 기존 하둡의 맵리듀스 성능을 다음과 같이 모델링 할 수 있다.

$$t_{hadoop} = (\frac{S_{buf}}{V_{tran}} + \frac{S_{buf}}{V_{map}}) * \frac{S_{map}}{S_{buf}} \quad (1)$$

또한 데이터 프리패칭이 적용된 맵리듀스의 성능은 다음과 같다.

$$t_{prefetch} = \frac{S_{buf}}{V_{tran}} + \frac{S_{buf}}{V'_{map}} + (\frac{S_{map} - S_{buf}}{\min(V'_{map}, V_{tran})}) \quad (2)$$

위 모델링에서 V'_{map}, V_{tran} 두 개의 값 사이에 작은 값은 택하는 이유는 데이터 전송시간이 데이터 처리시간보다 빠를 경우를 고려한 것이다. 하지만 대부분의 경우 데이터 처리시간보다 데이터 전송시간이 문제가 되므로, 본 논문에서는 전송 시간이 짧은 경우만을 고려하였다. 또한 V'_{map} 과 V_{map} 는 전송 버퍼에 사용되는 작은 메모리 공간을 제외하면 리소스의 차이가 없으므로, 성능 차이는 무시해도 될 수준이다. 이를 고려한 모델링은 다음과 같다.

$$t_{prefetch} = \frac{S_{buf}}{V_{tran}} + \frac{S_{buf}}{V_{map}} + (\frac{S_{map} - S_{buf}}{V_{tran}}) \quad (3)$$

프리패칭 맵리듀스 모델링에서의 $\frac{S_{buf}}{V_{tran}}$ 은 맵태스크가 최초로 입력데이터를 처리하기 위한 데이터 전송이다. 이 부분을 스트리밍 기법이 적용된 전송으로 표현 할 경우, 데이터 처리까지의 대기시간은 최소한의 처리를 할 수 있는 데이터가 전송 완료되는 시점인 $\frac{S_{streambuf}}{V_{tran}}$ 가 된다. 또한, 스트리밍 기법을 적용하더라도 전체 입력 파일 사이즈는 변하지 않으므로 최초의 파일 전송 외에 나머지 파일이 전송되는데 소모되는 시간은 기존의 프리패칭 기법에서 모델링 한 $\frac{S_{map} - S_{buf}}{V_{tran}}$ 이며, 이와 동일하게 파일 처리 속도 또한 $\frac{S_{buf}}{V_{map}}$ 로 나타 낼 수 있다. 따라서 본 논문에서 제안 하는 프리패칭 및 스트리밍 기법이 적용된 하둡 맵 리듀스의 작업 처리 시간의 모델링은 다음과 같다.

$$t_{improved} = \frac{S_{streambuf}}{V_{tran}} + \frac{S_{buf}}{V_{map}} + (\frac{S_{map} - S_{buf}}{V_{tran}}) \quad (4)$$

프리패칭 및 스트리밍 기법이 적용된 맵 리듀스가 기존의 하둡 맵 리듀스보다 얼마나 성능이 향상 되었는지 측정하기 위한 수식은 다음과 같다.

$$\triangle t = (\frac{S_{buf}}{V_{tran}} + \frac{S_{buf}}{V_{map}}) * \frac{S_{map}}{S_{buf}} - (\frac{S_{streambuf}}{V_{tran}} + \frac{S_{buf}}{V_{map}} + (\frac{S_{map} - S_{buf}}{V_{tran}})) \quad (5)$$

이 수식을 이용하여 기존 하둡 맵리듀스 와의 성능 비교를 위한 인자 값들을 표 2 와 같이 설정 하였다.

표 2. 기존 맵리듀스와 향상된 맵 리듀스의 성능 비교를 위한 인자 값 설정

Table 2. Values for Compare Original Mapreduce and improved Mapredcue

값	Case 1	Case 2	Case 3
V_{tran}	1,024	1,024	1,024
V_{map}	1,024	2,048	2,048
$S_{streambuf}$	1,024	512	10
S_{buf}	1,024	1,024	1,024
S_{map}	102,400	102,400	102,400

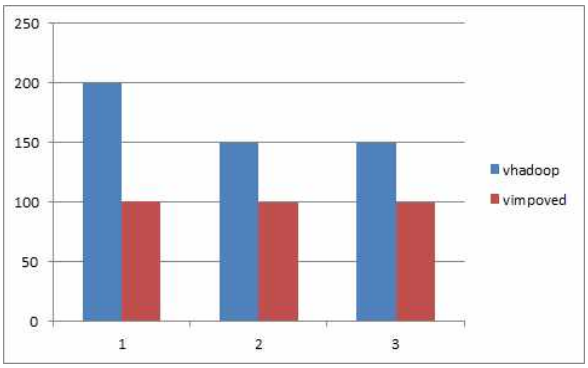

그림 5. 기존 맵리듀스와 향상된 맵리듀스의 성능비교

Fig. 5. Performance Comparing between Original Mapreduce and Improved Mapreduce

그림 5 는 기존 하둡 맵리듀스에 비해 프리패칭과 스트리밍 기법이 적용된 하둡이 더 좋은 성능을 나타냄을 보여준다. 또한, 맵태스크 처리 성능과 데이터 전송 성능의 차이가 크지 않을수록 프리패칭에 의한 성능 향상의 폭이 더 커짐을 알 수 있다.

기존의 프리패칭만 적용된 시스템과 스트리밍 기법을 추가로 적용한 시스템의 성능을 비교하기 위한 설정은 아래의 표 3 과 같다.

표 3. 프리패칭이 적용된 맵리듀스와 향상된 맵리듀스의 성능비교를 위한 인자 값 설정

Table 3. Values for Comparing Prefetching Mapreduce and Improved Mapreduce

값	Case 1	Case 2	Case 3
V_{tran}	1,024	1,024	1,024
V_{map}	2,048	2,048	2,048
$S_{streambuf}$	1,024	512	10
S_{buf}	1,024	1,024	1,024
S_{map}	102,400	102,400	102,400

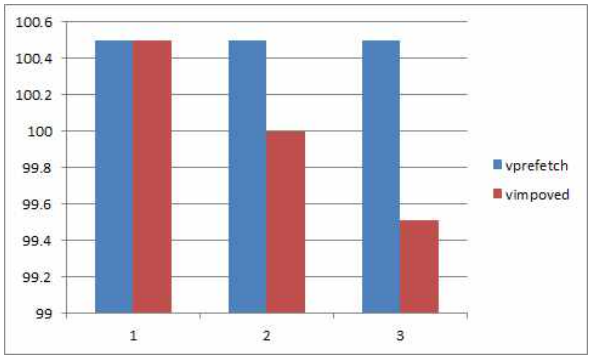

그림 6. 프리패칭이 적용된 맵리듀스와 향상된 맵리듀스의 성능비교
Fig. 6. Performance Comparing between Prefetching Mapreduce and Improved Mapreduce

스트리밍 기법 적용 시 스트리밍 버퍼의 사이즈가 작을수록 전송대기 시간이 감소한다. 그림 6의 결과는 프리패칭만 적용된 맵리듀스에 비해 스트리밍 기법이 추가 적용된 맵리듀스의 성능이 향상됨을 보여준다.

IV. 결 론

본 논문에서는 하둡 맵리듀스의 데이터 전송에 따른 성능 저하 문제를 해결하고자 프리패칭 기법 과 스트리밍 기법을 제안하였다. 맵 단계에서 프리패칭 기법을 적용함으로써 데이터 처리와 전송을 병렬로 처리 하여 데이터 전송 대기 시간을 줄일 수 있었고, 프리패칭 기법을 적용 할 수 없는 최초 데이터 전송에 의한 작업 처리 대기 시간을 줄이기 위해 스트리밍 기법을 추가 적용하였다. 또한 결과물의 성능을 측정하기 위해 수학적인 모델링을 하였으며, 성능 측정결과 기존의 하둡 맵리듀스 및 프리패칭 기법만 적용된 맵리듀스 보다 스트리밍 기법이 추가 적용된 맵리듀스의 성능이 향상되었음을 확인 할 수 있었다.

향후 스트리밍 기법의 적용 시 작은 데이터 전송 사이즈에 의한 패킷 오버헤드 문제를 해결하기 위해, 패킷 당 전송 데이터의 크기 결정 정책에 대한 연구를 진행할 예정이다.

ACKNOWLEDGMENT

본 연구는 BK21+사업, 한국연구재단 기초연구사업 (2012R1A1A2040257), (2013R1A1A2060398), 삼성전자(S-2014-0700-000), 미래창조과학부 및 정보통신기술연구진흥센터의 정보통신·방송 연구개발사업 (1391105003)의 일환으로 수행하였음.

참고문헌

[1] J. Dean and S. Ghemawat, “MapReduce: Simplified Data Processing on Large Clusters”, Communications of the ACM, vol. 51, no. 1, (2008).

[2] D. Jiang, B. Chin, L. Shi and S. Wu, “The performance of MapReduce: an in-depth study”, Proceedings of the VLDB Endowment. vol. 3, no. 1, (2010).

[3] J. Dean and S. Ghemawat, “Mapreduce: a flexible data processing tool”, Communications of the ACM, vol. 53, no. 1, (2010).

[4] K. Shvachko, H. Kuang, S. Radia and R. Chansler, “The Hadoop Distributed File System”, IEEE 26th Symposium on Mass Storage Systems and Technologies (MSST), (2010) May 3-7: Incline Village, USA.

[5] Tao Gu, Chuang Zuo, Qun Liao, Yulu Yang and Tao Li “Improving MapReduce Performance by Data Prefetching in Heterogeneous or Shared Environments “ International Journal of Grid and Distributed Computing Vol.6, No.5 (2013), pp.71-82

[6] Jiadong Wu and Bo Hong, "Improving MapReduce Performance by Streaming Input Data from Multiple Replicas" Cloud Computing Technology and Science (CloudCom), 2013 IEEE 5th International Conference Vol 1, (2013) pp.623 - 630