Smart Media Journal / Vol.8, No.4 / ISSN:2287-1322
http://dx.doi.org/10.30693/SMJ.2019.8.4.46

하둡 클러스터의 대역폭을 고려한 압축 데이터 전송 및 저장 기법

(Data Transmitting and Storing Scheme based on Bandwidth in Hadoop Cluster)

김용민*, 김희진*, 김영관*, 홍지만**

(Youngmin Kim, Heejin Kim, Younggwan Kim, Jiman Hong)

요 약

산업 현장과 공공 기관에서 생성 및 수집되는 데이터의 크기가 빠르게 증가하고 있다. 기존의 데이터 처리 서버는 스케일 업 방식으로 성능을 높여 증가하는 데이터를 처리하였다. 그러나 데이터의 생성 속도가 폭증하는 빅데이터 시대에는 기존 방식의 서버로는 데이터 처리에 한계가 있다. 이러한 한계를 극복하기 위해 스케일 아웃 방식으로 데이터를 분산 처리하는 분산 클러스터 컴퓨팅 시스템이 등장하게 되었다. 그러나 분산 클러스터 컴퓨팅 시스템은 데이터를 분산 처리하기 때문에 네트워크 대역폭을 비효율적으로 사용할 경우 클러스터 전체의 성능을 하락시킬 수 있다.

본 논문에서는 네트워크 대역폭을 고려하여 하둡 클러스터에서 데이터 전송 시 데이터를 압축 전송하는 기법을 제안한다. 제안 기법은 네트워크 대역폭과 압축 알고리즘의 특징을 고려하여 최적의 압축 전송 기법을 선정 후 전송한다. 실험 결과는 제안 기법을 사용할 경우 데이터 전송 시간과 크기를 감소시킨 것을 보여준다.

■ 중심어 : 하둡 ; 하둡 클러스터 ; 네트워크 대역폭 ; 데이터 전송 ; 압축

Abstract

The size of data generated and collected at industrial sites or in public institutions is growing rapidly. The existing data processing server often handles the increasing data by increasing the performance by scaling up. However, in the big data era, when the speed of data generation is exploding, there is a limit to data processing with a conventional server. To overcome such limitations, a distributed cluster computing system has been introduced that distributes data in a scale-out manner. However, because distributed cluster computing systems distribute data, inefficient use of network bandwidth can degrade the performance of the cluster as a whole.

In this paper, we propose a scheme that compresses data when transmitting data in a Hadoop cluster considering network bandwidth. The proposed scheme considers the network bandwidth and the characteristics of the compression algorithm and selects the optimal compression transmission scheme before transmission. Experimental results show that the proposed scheme reduces data transfer time and size.

■ keywords : Hadoop ; Hadoop Cluster ; Network Bandwidth ; Data Transmission ; Compression

Ⅰ. 서 론

정보통신 기술의 발전하면서 다양한 스마트 기기들이 사용되고, 이들 기기들이 생산하는 데이터가 빠르게 증가하고 있다[1]. 또한, YouTube, Facebook과 같은 SNS가 성장하면서 생성되는 데이터가 실시간으로 공유되고 있다. Wikibon의 2018 Big Data Analytics Trends and Forecast에 따르면 빅데이터 시장 규모는 2018년에 420억 달러에서 2027년 1,030억 달러에 이를 것으로 추정된다[2]. 그러나 하드웨어의 성능을 높여 데이터를 처리하는 기존 방법으로는 빅데이터 처리에 제한이 있다. 이를 해결하기 위해 대용량의 데이터를 분산하여 처리하는 하둡 시스템이 등장하게 되었다[3].

하둡은 빅데이터를 처리할 때 병렬처리를 통해 여러 노드에 데이터를 분산하여 처리한다. 하둡은 마스터 / 슬레이브 구조로 이루어져 있다. 마스터 노드인 네임 노드는 파일의 메타 데이터를 저장하고, 슬레이브 노드인 데이터 노드가 실제 데이터를 블록 단위로 저장한다. 하둡은 단일 노드의 성능을 높여 수직적으

* 준회원, 숭실대학교 컴퓨터학과 석사과정
** 정회원, 숭실대학교 컴퓨터학부 교수
이 논문은 2017년도 정부(과학기술정보통신부)의 재원으로 한국연구재단-차세대정보 · 컴퓨팅기술개발사업의 지원을 받아 수행된 연구임(No. NRF-2017M3C4A7069432).
접수일자 : 2019년 12월 09일 게재확정일 : 2019년 12월 18일
교신저자 : 홍지만, e-mail : jiman@ssu.ac.kr

로 성능을 향상시키는 스케일 업(Scale-up) 방식이 아니라 여러 노드를 사용하여 수평적으로 확장하는 스케일 아웃(Scale-out) 방식으로 성능을 높인다. 하둡은 JVM에서 수행되어 플랫폼에 독립적이다. 또한 내고장성이 있어 저비용의 하드웨어로 클러스터를 구성할 수 있다.

하둡 잡의 수행에 영향을 주는 주요 자원으로는 CPU, 디스크, 메모리 및 네트워크 대역폭이 있다. 특정 노드에 워크로드가 증가하여 주요 자원 중 하나라도 부족하게 될 경우 하둡 잡의 수행 시간이 길어지게 된다. 특정 노드에서 CPU, 디스크, 메모리 자원이 부족할 경우 해당 노드에서 수행되는 태스크 수행 시간이 길어지게 된다. 그러나 네트워크 대역폭은 하둡 클러스터에서 공유되는 자원이다. 따라서 네트워크 대역폭이 부족할 경우 하둡 클러스터의 모든 노드의 성능에 영향을 주게 된다[4].

하둡은 고가용성 지원을 위해 데이터 쓰기 작업 시 데이터를 복제하여 저장한다. 또한, 데이터 노드에 문제가 발생할 경우 해당 노드에서 저장하고 있는 모든 데이터 블록의 복제본을 다른 노드에 저장하게 된다. 또한 하둡은 빅데이터 처리를 위해 설계되었으므로 사용하는 데이터의 크기가 다른 시스템보다 크다. 따라서 하둡에서 데이터 전송이 발생할 경우 네트워크 자원의 사용량은 급증하고, 이는 하둡 클러스터에서 수행되는 모든 작업에 영향을 준다.

데이터 압축은 길이가 길고 자주 나오는 문장을 짧게 치환하고, 높은 빈도의 문자에 적은 비트를 부여하는 방식으로 수행된다. 따라서 네트워크를 통해 데이터를 전송할 때, 데이터를 압축 후 전송할 경우 데이터의 크기를 줄일 수 있어 네트워크 대역폭 사용량을 줄일 수 있다[5]. 그러나 데이터를 압축 후 전송할 경우 압축 및 압축 해제 오버헤드가 발생하여 데이터 전송 시간에 영향을 줄 수 있다. 따라서 본 논문은 데이터 전송 시 네트워크 대역폭을 고려하여 압축 전송이 효율적일 때만 데이터를 압축 후 전송하여 네트워크 자원 사용량을 감소시키는 연구를 수행한다. 또한, 이를 통해 하둡 클러스터의 성능을 향상시키도록 한다.

데이터 압축 알고리즘에 따라 압축 및 압축 해제 시간, 압축비 등이 다르다[6]. 압축 알고리즘마다 특징이 다르기 때문에 네트워크 대역폭에 따른 적절한 압축 알고리즘이 다르다. 데이터 전송 비용 계산 시 네트워크 대역폭과 압축 알고리즘의 특징을 반영하여 압축 알고리즘을 선정한다. 데이터 압축 전송 시 파이프라인이 형성되므로, 데이터 전송 비용 계산 시 이를 반영한다.

본 논문의 구성은 다음과 같다. 2장에서는 하둡의 성능을 향상시키기 위해 수행된 관련 연구를 소개한다. 3장에서는 본 논문의 제안 기법을 소개하고, 4장에서는 본 논문을 결론짓고 향후 연구 방향을 제시한다.

Ⅱ. 관련 연구

노승준 등은 [7]에서 스트리밍 압축 기법을 통해 하둡 클러스터의 네트워크 사용량을 감소시킨 연구를 수행하였다. 제안 기법을 통해 기존 하둡의 쓰기 연산보다 네트워크 사용량을 최대 56% 감소시켰다. 그러나 해당 연구는 데이터 압축 시 파일 엔트로피, 데이터 압축비, 네트워크 대역폭 등을 고려하지 않아 기존 기법보다 46%의 오버헤드가 증가가 발생하였다.

Kritwara Rattanaopas 등은 데이터 압축 알고리즘에 따른 맵리듀스 성능을 비교하는 연구를 수행하였다[8]. Bzip2 압축 알고리즘을 사용하여 데이터를 압축할 경우 맵리듀스 성능의 감소는 미미하나 저장 공간을 60% 줄일 수 있었다. 그러나 해당 연구는 압축 해제 오버헤드만을 반영한 문제가 있다. 또한, 데이터 압축비와 노드의 성능 등을 고려하지 않았다.

Xinxin Fan 등은 하둡 클러스터의 스케줄링에 네트워크 대역폭을 도입하여 하둡 클러스터의 성능을 향상시키는 연구를 수행하였다[9]. 하둡에서 기본적으로 제공하는 DRF 알고리즘에 네트워크 대역폭을 리소스로 추가하였다. 네트워크 대역폭은 LTC를 통해 격리하였다. LTC를 통해 리소스 할당을 유연하게 하였고, 대역폭의 최소 성능을 보장하였다. 해당 연구는 네트워크 자원을 스케줄링에 고려하여 하둡 클러스터의 성능을 개선했으나, 전체 네트워크 대역폭 사용량을 감소시키지 못하였다.

Yanfei Guo 등은 허듑의 맵리듀스 태스크 수행 과정 중 리듀스 단계에서 셔플 과정을 분리하여 맵리듀스 태스크의 성능을 높이는 연구를 수행했다[10]. 셔플 과정 분리를 통해 리듀스 태스크를 미리 수행하였고, 이를 통해 하둡의 성능을 향상시켰다. 해당 연구는 맵리듀스 태스크에서 딜레이를 제거하여 CPU 자원의 효율적인 사용을 통해 수행 시간을 줄였으나, 네트워크 대역폭의 사용량을 낮추지 못하였다.

Ⅲ. 본 론

1. 하둡 클러스터의 대역폭과 데이터 압축

가. 하둡 클러스터 비용 모델

하둡 클러스터의 데이터 압축 전송 비용을 측정을 위해 Herodotos Herodotou가 [11]에서 제안한 하둡 비용 모델을 사용한다. 해당 논문에서 제안한 비용 모델은 CPU 비용, I/O 비용, 데이터 전송 비용의 합으로 계산한다. 본 논문은 해당 모델을 수정 후 사용한다.

$$IOCost = IOCost_{Read} + IOCost_{Write} \quad (1)$$

$$IOCost_{Read} = \frac{Len}{Ratio} \times readCost_b \quad (2)$$

$$IOCost_{Write} = \frac{Len}{Ratio} \times writeCost_b \quad (3)$$

수식 (1)~(3)은 I/O 비용을 계산하는 식이다. I/O 비용은 디스크 읽기 비용과 쓰기 비용의 합으로, I/O 작업 시 발생하는 비용이다. 읽기 비용 및 쓰기 비용은 데이터를 메모리로 로드하거나 디스크로 쓸 때 발생하는 비용이다. 읽기 및 쓰기 비용은 데이터 크기와 단위 비트 당 압축 및 압축 해제 비용에 비례한다. 따라서 읽기 및 쓰기 비용은 데이터 크기와 단위 비트 당 압축 및 압축 해제 비용에 비례하고, 압축비에 반비례한다. 단위 비트 당 압축 및 압축 해제 비용은 하드웨어 성능과 압축 알고리즘에 따라 다른 값을 갖는다.

$$CPUCost = CPUCost_{Comp} + CPUCost_{Uncomp} \quad (4)$$

$$CPUCost_{Comp} = Len \times compCPUCost_b \quad (5)$$

$$CPUCost_{Uncomp} = \frac{Len}{Ratio} \times uncompCPUCost_b \quad (6)$$

수식 (4)~(6)은 CPU 비용을 계산하는 식이다. CPU 비용은 압축 비용과 압축 해제 비용의 합으로, CPU 사용 시 발생하는 비용이다. 압축 및 압축 해제 비용은 데이터 크기와 단위 비트 당 압축 및 압축 해제 비용에 비례한다. 압축 비용의 경우 비압축 데이터를 대상으로 하여 압축비를 수식에서 제외한다. 단위 비트 당 압축 및 압축 해제 비용은 하드웨어 성능과 압축 알고리즘, 데이터의 특징에 따라 다른 값을 갖는다.

$$NETCost_{Job} = \frac{Len}{Ratio} \times netCost_b \quad (7)$$

수식 (7)은 데이터 전송 비용을 계산하는 식이다. 데이터 전송 비용은 데이터 크기와 단위 비트 당 데이터 전송 비용으로 계산할 수 있다. 따라서 데이터 전송 비용은 데이터 크기와 단위 비트 당 데이터 전송 비용에 비례하고, 압축비에 반비례한다. 단위 비트 당 데이터 전송 비용은 네트워크 대여폭을 측정하여 계산할 수 있다.

나. 데이터 압축과 네트워크 대역폭

데이터 전송 시 데이터를 압축 후 전송할 경우 데이터 전송 비용은 디스크 비용, 네트워크 비용, 압축 비용, 데이터 길이 및 압축비로 계산할 수 있다. 따라서 데이터 전송 비용 수식에 값을 대입하면 압축 알고리즘별 데이터 전송 비용을 계산할 수 있다. 알고리즘에 따른 데이터 전송 비용을 계산할 경우 최저 비용의 데이터 전송 비용을 계산할 수 있고, 해당 알고리즘으로 데이터 전송 시 하둡 클러스터의 성능을 높일 수 있다.

데이터 압축 전송 비용 중 데이터 크기는 클라이언트 노드에서 데이터 노드로 전송할 데이터의 크기다. 디스크 읽기 비용은 동일한 하드웨어에서 같은 데이터를 사용하므로 동일한 값을 갖는다. 데이터 노드는 압축된 데이터를 수신할 경우 태스크 수행 시 압축 해제 오버헤드를 제거하기 위해 압축 해제 후 저장을 한다. 따라서 데이터 쓰기 비용은 알고리즘에 상관없이 동일한 값을 갖는다. 데이터 크기는 같은 데이터를 사용할 경우 동일한 크기를 갖는다. 따라서 압축 알고리즘에 따른 데이터 압축 전송 비용 비교 시 동일한 비용을 제거할 경우 데이터 전송 식은 수식 (8)을 통해 나타낼 수 있다.

$$Cost = compCPUCost_b + \frac{networkCost_b}{Ratio} + \frac{uncompCPUCost_b}{Ratio} \quad (8)$$

압축 알고리즘에 따른 데이터 압축 전송 비용을 비교할 경우 압축 데이터 전송 비용은 네트워크 대역폭과 압축 알고리즘에 의해 결정된다. 수식 (8)을 통해 네트워크 대역폭에 따른 최저 비용을 계산할 수 있고, 이를 통해 최적의 압축 알고리즘을 선정할 수 있다. 그림 1은 압축 알고리즘과 네트워크 대역폭에 따른 데이터 전송 시간을 보여준다.

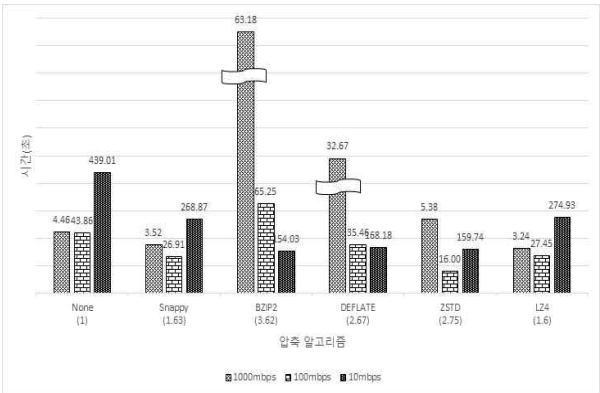

그림 1. 압축 알고리즘과 네트워크 대역폭에 따른 압축 전송 시간

그림 1은 네트워크 대역폭에 따라 최적의 압축 알고리즘이 다른 것을 확인할 수 있다. 네트워크 대역폭이 작은 경우 압축률이 높은 BZIP2 압축 알고리즘을 사용할 때 가장 좋은 성능을 보인다. 네트워크 대역폭이 중간인 경우 압축 속도와 압축률이 중간인 ZSTD 압축 알고리즘을 사용할 때 가장 좋은 성능을 보인다. 네트워크 대역폭이 클 경우 압축 속도가 가장 빠른 LZ4 압축 알고리즘이 가장 좋은 성능을 보인다. 따라서 데이터를 압축 후 전송할 때 네트워크 대역폭에 따라 최소 비용을 갖는 압축 알고리즘을 선정할 경우 데이터를 최저 비용으로 전송할 수 있다.

다. 데이터 압축 전송 파이프라인

클라이언트 노드가 데이터 노드로 데이터를 압축 후 전송할 경우 데이터 전송 비용은 압축 비용, 데이터 전송 비용, 압축 해제 비용의 합으로 계산할 수 있다. 그러나 압축 과정, 데이터 전송 과정, 압축 해제 과정이 모든 다른 하드웨어에서 수행될 경우 각 단계는 중첩되어 실행된다. 따라서 데이터를 압축 후 전송할 때 여러 청크로 쪼개서 전송할 경우, 압축 과정, 데이터 전송 과정, 압축 해제 과정은 파이프라인이 형성되어 병렬적으로 수행된다.

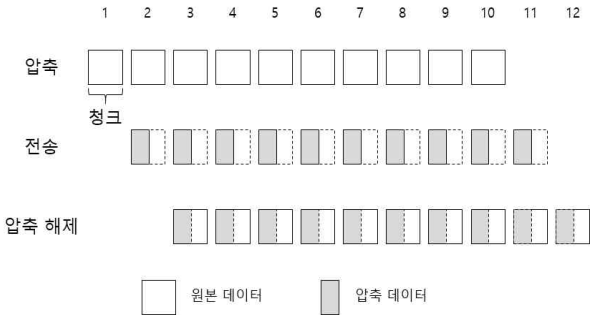

그림 2. 압축 데이터 전송 파이프라인

그림 2는 데이터를 여러 개의 청크로 나누어 보낼 경우 파이프라인이 형성되는 것을 보여준다. 1번 단계에서는 첫 번째 청크의 압축 과정만 존재한다. 2번 단계에서는 압축과 전송이 동시에 발생한다. 3번부터 10번 단계는 데이터 압축, 전송, 압축 해제 과정이 동시에 발생한다. 전송 청크의 수가 많을 경우 초기와 마지막 단계를 무시할 수 있다. 따라서 청크 하나의 압축, 전송, 압축 해제 비용의 합은 압축, 전송, 압축 해제 비용 중 가장 큰 값이 된다.

lzbench[12]의 벤치마크 결과에 따르면 특정 압축 알고리즘을 제외하고는 압축 속도가 압축 해제 속도보다 빠르다. 또한 본 논문에서 사용하는 압축 알고리즘은 압축 해제 속도가 압축 속도보다 약 3배정도 빠르다. 따라서 청크의 전송 비용에서 압축 해제 비용을 제거할 수 있다. 이를 수식으로 나타내면 아래와 같다.

$$Cost_{Uncomp} = netCost_{c} \times n$$

$$Cost_{Comp} = \sum_{i=1}^{n} \max(compCPUCost_{c}, \frac{netCost_{c}}{Ratio})$$

위 수식은 전체 데이터를 전송할 때 발생하는 비용이다. 데이터 전송 시 압축 시점을 각 청크의 전송 시점으로 볼 경우 비압축 전송 비용은 $netCost_{c}$가 되고, 압축 전송 비용은 $\max(compCPUCost_{c}, \frac{netCost_{c}}{Ratio})$가 된다. 따라서 데이터 전송 시 두 비용을 비교하여 압축 여부를 결정하고, 최적의 압축 알고리즘을 선정하여 데이터를 전송한다.

2. 구현

본 논문은 네트워크 대역폭을 기반으로 데이터를 압축 전송하는 기법을 제안한다. 하둡 클러스터에서 클러스터가 데이터 노드로 쓰기 작업 시 네트워크 대역폭과 압축 알고리즘의 특징을 고려하여 데이터 압축 전송 비용을 계산 후 전송한다. 그러나 데이터 쓰기 작업 초기에는 압축 전송 비용을 계산할 때 필요한 값이 없다. 따라서 압축 전송 초기에는 일부 패킷을 압축 알고리즘별로 압축 후 전송하여 필요한 값을 구한다. 그 다음 필요한 값을 모두 구한 후 데이터 전송 시 압축 알고리즘에 따른 최저 비용을 계산 후 데이터를 압축 전송한다.

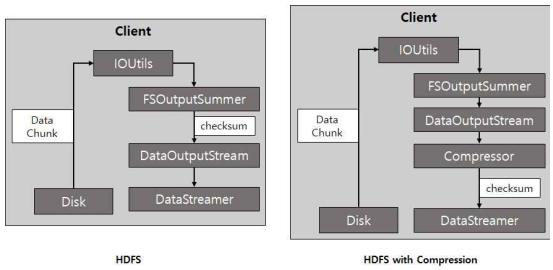

그림 3. 기존 HDFS의 쓰기 과정과 수정된 쓰기 과정

그림 3은 기존 하둡의 클러스터에서 쓰기 작업의 흐름과 본 논문에서 제안하는 기법을 적용한 쓰기 작업의 흐름을 보여준다. 기존 기법의 경우 'IOUtils' 클래스로 데이터를 읽어 'FSOutputSummer' 클래스로 전송한다. 'FSOutputSummer' 클래스는 전송 받은 데이터를 설정된 크기 단위로 체크섬을 생성 후 'DataOutputStream'으로 전송한다. 'DataOutputStream' 클래스는 전달 받은 데이터와 체크섬을 저장 후 설정된 크기가 되면 패킷으로 만들어 데이터 노드로 전송한다.

수정된 쓰기 기법의 경우 데이터 노드로 패킷 전송 시 네트워크 대역폭과 압축 알고리즘의 특징을 고려하여 최저 비용의 압축 알고리즘으로 압축 후 데이터를 전송한다. 데이터를 압축 후 체크섬을 생성해야 하므로 체크섬의 생성 과정을 뒤로 미룬다. 디스크에서 데이터를 충분히 읽었을 경우 이를 압축 후 체크섬을 생성하여 데이터 노드로 전송한다. 위 과정을 수행하기 위해 'Compressor' 클래스를 추가한다. 'Compressor' 클래스는 데이터 압축 및 체크섬 생성 외에도 최저 비용의 압축 알고리즘을 선정하는 역할도 수행한다.

3. 실험 및 평가

가. 실험

본 논문에서 제안한 엔트로피 및 대역폭을 고려한 데이터 압축 전송 및 저장 기법에 대해 다양한 환경에서 실험을 수행한다. 실험 결과를 기존의 하둡 파일 시스템과 단일 압축 알고리즘을 통해 압축 전송한 기법과 비교를 한다. 데이터 전송 시 로그를 생성하게 하여 수행 시간을 측정하였고, 네트워크 대역폭을 변경시키며 수행 시간의 변화를 확인하였다.

(1) 실험 환경

제안 기법의 성능을 평가하기 위해 실험 환경을 표 1과같이 구축하였다. 4대의 PC를 1개의 네임 노드와 3개에 데이터 노드로 구성하였다. 모든 PC는 우분투 18.0.4 버전에 하둡 3.1.2 버전을 사용하였다.

표 1. 하둡 클러스터 실행 환경

구분	H/W	
네임 노드	CPU	Intel(R) Core(TM) i5-4670 CPU @ 3.40GHz
	메모리	8GB
	디스크	Samsung 840 Series PRO 512GB
데이터 노드1	CPU	Intel(R) Core(TM) i5-3330 CPU @ 3.00GHz
	메모리	8GB
	디스크	WesternDigital 1TB WD10EZEX
데이터 노드2	CPU	Intel(R) Core(TM) i5-4670 CPU @ 3.40GHz
	메모리	8GB
	디스크	Samsung 840 Series 120GB Seagate 2TB ST2000DM001
데이터 노드3	CPU	Intel(R) Pentium G3440 CPU @ 3.30Ghz
	메모리	8GB
	디스크	Seagate 1TB ST1000DM003

(2) 실험 방법

본 논문에서 제안한 기법을 평가하기 위해 네트워크 대역폭을 변경하여 실험을 진행한다. 네트워크 대역폭은 LTC(Linux Traffic Controll)를 사용하여 변경하였고, 이를 통해 여러 노드가 네트워크 대역폭을 사용하는 상황을 구현하였다. 네트워크 대역폭을 무작위 또는 규칙적으로 변경시키며 제안 기법의 성능을 평가한다. 실험 데이터로는 Gutenberg 데이터 셋을 사용하였고, 전송 시간은 3번을 측정하여 평균을 내었다[11].

나. 실험

제안 기법의 성능을 평가하기 위해 다른 압축 알고리즘과 제안 기법을 비교한다. 네트워크 대역폭의 변화 방식에 따른 각 압축 알고리즘의 데이터 전송 시간과 크기를 측정하였다.

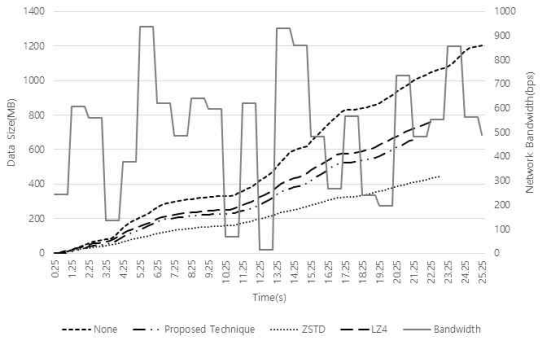

그림 4. 대역폭의 무작위 변화와 데이터 전송 시간 및 크기

그림 4는 네트워크 대역폭이 1Mbps~1,000Mbps의 값 중 무작위로 변할 때 압축 알고리즘에 따른 데이터 전송 시간과 크기를 보여준다. 대역폭의 변경 주기는 1초이다. 데이터를 비압축으로 전송할 경우 전송 시간이 가장 느렸고, 제안 기법을 사용할 경우 전송 시간이 가장 빨랐다. 전송 데이터의 크기는 ZSTD 압축 알고리즘이 가장 적었다.

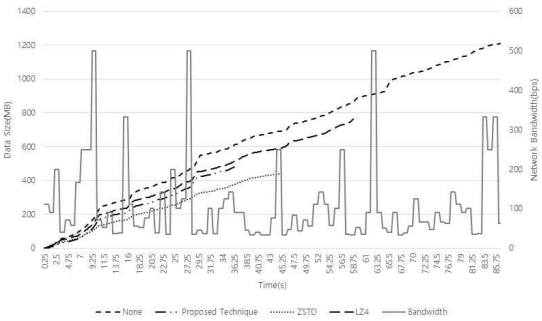

그림 5. 네트워크 참여 노드의 무작위 변화와 데이터 전송 시간 및 크기

그림 5는 최대 네트워크 대역폭이 1,000Mbps인 네트워크에 노드가 무작위로 접속했을 때 압축 알고리즘에 따른 데이터 전송 시간과 크기를 보여준다. 노드의 최대 수는 30이고, 노드와 대역폭은 1초마다 변경된다. 데이터를 비압축으로 전송할 경우 전송 시간이 가장 느렸고, 제안 기법을 사용할 경우 전송 시간이 가장 빨랐다. 전송 데이터의 크기는 ZSTD 압축 알고리즘이 가장 적었다.

Ⅳ. 결 론

본 논문은 하둡 분산 파일 시스템에 네트워크 대역폭을 고려한 데이터 압축 정송 기법을 적용하여 하둡 클러스터의 성능을 높이는 연구를 수행하였다. 비압축 전송 비용과 개별 압축 알고리즘의 전송 비용을 계산 후 최적의 비용으로 데이터를 전송하였다.

실험을 통해 제안 기법, 비압축 전송, 단일 압축 알고리즘 압축 전송 기법을 비교하였다. 실헐 결과 제안 기법이 비압축 및 단일 압축 전송 기법보다 데이터를 빠르게 전송할 수 있었다. 특히 네트워크에 참여하는 노드의 수가 많을 경우 제안 기법의 성능이 다른 기법에 비해 우수하였다.

제안 기법과 ZSTD 압축 전송 기법을 비교할 경우 평균 네트워크의 속도가 높을 때 전송 시간은 비슷하였으나, ZSTD 알고리즘의 데이터 크기가 작았다. 따라서 제안 기법의 성능을 높이기 위해서는 데이터 전송 시간이 비슷할 경우 전송 데이터의 크기를 줄이는 연구가 필요하다.

REFERENCES

[1] 김남호, 노진헌, 정희자, “RFID/NFC 물류의 빅 데이터 처리를 위한 하둡 시스템의 설계,” *스마트미디어저널*, 제2권, 제3호, 47-53쪽, 2013년 9월

[2] https://wikibon.com/wikibons-2018-big-data-analytics-trends-forecast (accessed Nov., 22, 2019).

[3] 이영훈, 김용일, “Hadoop 클러스터에서 네임 노드와 데이터 노드가 빅 데이터처리 성능에 미치는 영향에 관한 연구,” *스마트미디어저널*, 제6권, 제3호, 68-74쪽, 2017년 9월

[4] M. Laurent and R. James, “Bandwidth Sharing: Objectives and Algorithms,” *IEEE/ACM Transactions on Networking*, vol. 10, pp. 320-328, 2002.

[5] J. Pane and L. Joe, “Making Better Use of Bandwidth, Data Compression and Network Management Technologies,” *Technical Report*, Santa Monica, 2005.

[6] R. Ehab, “PERFORMANCE EVALUATION OF DATA COMPRES-SION TECHNIQUES VERSUS DIFFERENT TYPES OF DATA,” *International Journalof Computer Science and Information Security*, vol. 11, no. 12, pp. 73-78, 2013.

[7] 노승준, 엄영익, “하둡 시스템의 네트워크 자원 사용량 감소를 위한 스트리밍 압축 기법,” *한국정보과학회*, 2018

[8] R. Kritwara and K. Sureerat, “Imporving Hadoop MapReduce Performance with Data Compression: A Study using Wordcount Job,” *Proc. of the 14th International Conference on Electrical Engineering/Electronics, Computer, Telecommunications and Information Technology (ECTI-CON)*, pp. 564-567, IEEE, 2017

[9] F. Xinxin, L. Bo, Z. Yuan, and Z. Tianning, “Adding network bandwidth resource management to Hadoop YARN,” *Proc. of the Seventh International Conference on Information Science and Technology(ICIST)*, pp. 444-449, 2017.

[10] Y. Guo, J. Rao, and X. Zhou, “iShuffle: Improving Hadoop Performance with Shuffle-on-Write,” *IEEE transactions on parallel and distributed systems*, 2016.

[11] H. Herodotos, “Hadoop Performance Models,” Technical Report CS-2011-05, Duke Computer Science, 2011.

[12] https://github.com/inikep/lzbench (accessed Nov., 22, 2019).

저 자 소 개

김용민(준회원)

2018년 숭실대학교 컴퓨터학부 학사 졸업.
2018년~현재 숭실대학교 컴퓨터학과 석사 재학

<주관심분야 : 시스템 소프트웨어, 운영체제>

김희진(준회원)

2019년 숭실대학교 컴퓨터학부 학사 졸업.
2019년~현재 숭실대학교 컴퓨터학과 석사 재학

<주관심분야 : 시스템 소프트웨어, 운영체제>

김영관(준회원)

2019년 숭실대학교 컴퓨터학부 학사 졸업.
2019년~현재 숭실대학교 컴퓨터학과 석사 재학

<주관심분야 : 시스템 소프트웨어, 운영체제>

홍지만(종신회원)

2003년 서울대학교 컴퓨터공학과 박사 졸업
2004년~2007년 광운대학교 컴퓨터공학과 교수
2007년~현재 숭실대학교 컴퓨터학부 교수

<주관심분야 : 시스템 소프트웨어, 운영체제>