클라우드 컴퓨팅을 위한 분산 파일 시스템에서의 파일 병합 기법

이동우, 김정한, 엄영익
성균관대학교 정보통신공학부
e-mail : {cowsboys, gtgkjh, yieom}@ece.skku.ac.kr

Merging Files on Distribute File Systems for Cloud Computing

Dongwoo Lee, Junghan Kim, and Young Ik Eom
School of Information and Communication Engineering, Sungkyunkwan Univ.

요 약

최근 IT산업의 화두인 클라우드 컴퓨팅에서, HDFS는 널리 사용되고 있는 분산 파일 시스템이다. HDFS는 분산된 데이터의 저장과 검색의 장점이 있는 반면, 대용량 파일처리를 목적으로 설계되었기 때문에 실시간 파일처리와 저용량 데이터 처리에 비효율적이다. 본 논문에서는 이러한 문제를 해결하기 위해 HDFS의 파일 처리 과정을 개선하여 저용량 파일 처리를 향상시키는 방법을 제안한다. 본 기법은 데이터 블록에 저용량 파일들을 병합함으로써 데이터 처리의 효율성을 높이는 결과를 보였다.

1. 서론

하드웨어 기술의 발달과 통신기술의 향상으로 인터넷을 통해 IT자원을 온디맨드로 빌려 사용하는 클라우드 컴퓨팅은 차세대 IT 산업의 핵심 분야로 주목받고 있다[1].

컴퓨터 환경에서 데이터는 매우 중요한 요소이다. 우리가 사용하는 PC에도 수십, 수백 기가바이트 단위의 데이터들이 저장되어 있다. 특히 클라우드 컴퓨팅에서는 여러 사용자에 의해 막대한 양의 데이터가 다루어지게 된다. 그러므로 이 데이터를 수많은 서버들에 분산 저장하고 관리하는 분산 데이터 저장 기술은 클라우드 컴퓨팅의 핵심 기술이라고 할 수 있다.

오픈소스 분산 파일시스템인 HDFS는 최근 클라우드 컴퓨팅에서 주목 받는 파일 시스템이다. HDFS는 많은 노드의 로컬 파일 시스템을 묶어 하나의 대용량 단일 파일 시스템 환경을 제공한다. 또한 여러 데이터노드를 이용한 분산저장 기능을 제공하기 때문에 데이터 집중에 의한 특정 노드의 과부하를 방지 할 수 있다[2].

HDFS는 대용량 데이터 분산 처리 플랫폼인 Hadoop을 위해 고안되었다[3]. Hadoop에서는 일괄처리 방식을 사용하기 때문에 파일을 실시간으로 처리하는 데는 적합하지 않다. 또한 큰 용량의 파일을 효율적으로 처리하기 위해 수십 메가 단위의 큰 데이터블록 사용하기 때문에, 저 용량 데이터의 저장 시 저장 공간이 낭비되는 문제가 있다.

본 논문에서는 이러한 문제를 해결하고자 HDFS에 파일 병합 기법을 적용하였다. 저 용량 파일을 데이터블록 크기로 병합함으로써 데이터 처리의 효율성이 향상되었음을 확인하였다.

2. 관련 연구

2.1 클라우드 컴퓨팅

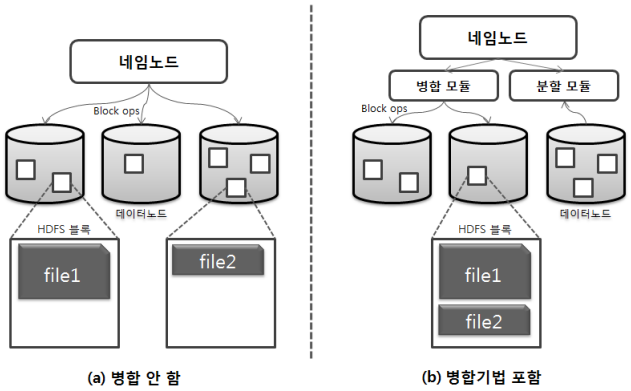
제한된 물리적인 인프라를 활용하여, 클라우드 환경에서 서비스 제공자는 다수의 사용자들이 자신만을 위한 IT 인프라가 구축되어 있는 것과 같은 환경을 제공한다. 서비스 제공자는 이를 위해 여러 곳에 분산되어 있는 물리적 인프라를 가상화하여 가상의 자원 풀(Resource Pool)을 구축하고 사용자의 요구에 따라 이러한 자원들을 효율적으로 배분한다[4]. 클라우드 컴퓨팅 서버는 주로 여러 서버를 묶어 클러스터링 환경을 구축한다. 이러한 클러스터링 환경에서 뛰어난 성능을 발휘하는 HDFS나 GFS와 같은 분산 파일 시스템이 클라우드 컴퓨팅을 위한 파일 시스템으로 주목받고 있다.

2.2 분산 파일시스템

오픈 소스 분산 처리 플랫폼인 Hadoop은 분산 컴퓨팅을 위한 신뢰성과 확장성을 갖춘 플랫폼을 제공한다. HDFS는 Hadoop 플랫폼을 지원하기 위해서 개발된 분산 파일 시스템이다. HDFS는 현재 클라우드 컴퓨팅의 기반이 되는 분산 파일 시스템으로 가장 널리 사용 되고 있다[5]. HDFS는 네임스페이스를 관리하고 클라이언트의 파일에 대한 접근을 통제하는 단일 네임노드와 데이터 저장소를 관리하는 수많은 데이터노드로 구성되며, 주로 Linux기반으로 동작하는 저비용 서버에서 구동되도록 설계되었다. HDFS는 64메가바이트 크기의 블록들로 파일들을 관리하며, 블록은 각각의 데이터노드들에 분산되어 저장된다.

3. 파일 병합 기법

앞에서 언급한 바와 같이, HDFS는 Hadoop 플랫폼을 지원하기 위한 분산 파일 시스템이다. 그로인해 저용량 파일의 잦은 입출력 처리에 비효율적인 문제점이 있다. 네트워크를 통해 처리되는 분산 파일시스템의 특징 때문에 잦은 파일 입출력은 네트워크 사용량이 증가된다는 문제가 있다. HDFS는 기본적으로 대용량 파일을 효율적으로 지원하기 위해 64MB의 블록을 사용한다. 즉 파일 크기가 1MB미만의 저 용량 파일이라도 논리적으로는 64MB의 공간을 차지한다. 그 결과 여러 사용자가 클라우드 컴퓨팅 서비스를 제공받는 경우 네트워크 대역폭과 저장 공간의 낭비를 초래한다. 따라서 본 논문에서는 사용자가 파일 입출력 처리를 할 때 낭비되는 시스템 자원을 최소화하기위해 저용량 파일을 병합하는 방법을 제안한다.

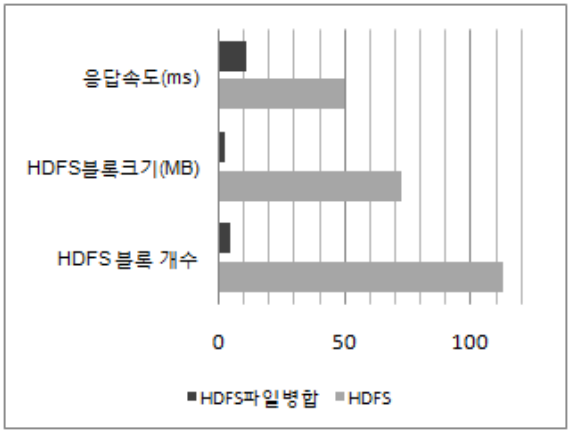


(그림 1) HDFS 블록 구조 비교

그림1(a)와 같이 기존 HDFS에서는 하나의 저 용량 파일 당 하나의 HDFS 블록이 필요하였다. 그러나 본 제안한 기법에서는 그림1(b)와 같이 64MB내에서 저 용량 파일들을 병합하여 하나의 블록에 여러 파일이 존재 하도록 했다. 그리고 파일의 사용이 모두 끝나 파일이 닫힐 경우에 즉시 HDFS의 네임노드에 메타데이터를 수정하지 않는다. 그 대신 64MB 블록이 가득차서 더 이상 파일을 병합할 수 없는 경우 이 블록을 실제로 저장한다. 또한 이와 같은 지연된 파일 갱신은 데이터 손실의 가능성이 있으므로, 일정 시간 마다 데이터블록을 자동으로 갱신하도록 하는 방법을 사용하였다.

4. 실험 결과 및 분석

본 논문에서는 제안 기법을 평가하기 대상 디렉토리내에 존재하는 파일들을 차례로 열고 닫는 동작을 시뮬레이션 하였다. 동작하는 동안 요구되는 데이터 블록의 개수 및 크기와 동작시간을 측정 하여 파일 병합기법을 사용하는 경우와 그렇지 않은 경우를 비교 하였다. 본 실험은 평균 파일 크기 10KB에서 100KB인 113개의 파일로 이루어진 /bin/ls 디렉토리를 대상으로 진행 하였다.



(그림 2) 파일 입출력 성능 비교

그림2와 같이 기존 HDFS에서는 하나의 저 용량 파일마다 HDFS 블록을 할당하기 때문에 7.2GB정도의 공간을 사용한다. 그러나 본 파일 병합기법을 적용한 결과에서는 64MB의 블록 한 개만을 사용한다. 또한 기존 HDFS는 전체 파일 닫을 때 총 7.2GB의 블록을 네임노드에서 갱신해야 하기 때문에 응답시간이 늦어진다. 그러나 병합기법을 적용했을 때는 최종적으로 마지막 파일이 닫힐 때 블록을 단 한번만 갱신하기 때문에 응답속도가 빠른 결과를 보였다.

5. 결론

본 논문에서는 파일 병합 기법을 통해 클라우드 컴퓨팅 환경에서 분산 파일시스템의 효율성을 향상시키는 방법을 제안하였다. 그러나 파일들의 크기가 커서 병합 할 수 없는 경우 오히려 동작 속도가 더 느려지는 문제를 확인하였다. 이는 개별 파일에 대한 병합 여부 검사 과정에서 발생하는 지연에 의한 것으로 판단된다. 차기 연구로는, 본 연구를 바탕으로 지연 문제를 해결하기 위한 병합 및 분할 과정의 최적화 연구를 진행 할 것이다.

참고문헌

[1] Rajkumar Buyya, Chee Shin Yeo, and Srikumar Venugopal."Market-oriented cloud computing: Vision, hype, and reality for delivering it services as computing utilities." 2008

[2] D. Borthakur, "The Hadoop Distributed File System: Architecture and Design," Hadoop Website, 2007

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

[4] M. Armbrust, A. Fox, R. Griffith et al., "Above the Clouds: A Berkeley View of Cloud Computing," Unversity of California, Berkeley, Berkeley, CA, 2009.

[5] 민영수, 김홍연, 김영균 "클라우드 컴퓨팅을 위한 분산 파일 시스템 기술" *한국정보과학회지*, 27(5), 86 (2009)