저장 공간의 효율성을 위한 분산 파일 시스템의 중복 데이터 제거 기법

김정훈, 임병홍, 엄영익
성균관대학교 정보통신공학부
e-mail:{myhuni20, dd8562, yieom}@ece.skku.ac.kr

Eliminating Redundant Data for Storage Efficiency on Distributed File Systems

Jung Hoon Kim, ByoungHong Lim, and Young Ik Eom
School of Information and Communication Engineering, Sungkyunkwan Univ.

요 약

최근 IT 분야의 키워드 중 하나인 클라우드 컴퓨팅에서, 분산 파일 시스템의 선택은 대용량의 데이터를 관리하기 위해 매우 중요하다. 오픈소스 분산 파일 시스템 중 하나인 HDFS는 데이터의 효율적인 저장과 검색의 장점을 통해 최근 널리 사용되고 있다. HDFS는 데이터를 3단계 중복저장을 통해 신뢰성을 보장한다. 그러나 이러한 중복저장은 데이터 저장의 효율성 저하의 문제점을 갖고 있다. 따라서 본 논문에서는 MD5 해쉬 기술을 적용한 중복 데이터 제거 기법을 제안한다. 본 기법은 시뮬레이션을 통해 저장 공간의 효율성을 향상의 결과를 확인하였다.

1. 서론

최근 IT 분야의 학계와 산업계에서 클라우드 컴퓨팅은 새로운 키워드로 부상하고 있다[1]. 아마존의 EC2(Elastic Compute Cloud)와 구글의 Apps는 이미 잘 알려진 클라우드 컴퓨팅의 예이다. 최근에는 Microsoft, IBM, HP, SUN 등 글로벌 기업들이 대거 참여하여 SaaS, PaaS, IaaS와 같은 다양한 형태의 클라우드 컴퓨팅 서비스를 선보이고 있다.

위와 같은 클라우드 컴퓨팅을 제공하기 위해서는 대규모 IT 인프라 구축이 우선적으로 필요하다. 무엇보다도 클라우드 컴퓨팅에서 사용되는 많은 양의 데이터를 저장하기 위한 분산 파일 시스템의 선택은 매우 중요하다. 2008년 초 제안된 HDFS는 MapReduce 알고리즘을 적용하여 분산된 노드의 효율적인 데이터 저장과 검색을 제공한다. 또한 높은 이식성과 내(耐)고장성을 보장한다. 이러한 장점을 통해서 HDFS는 최근 클라우드 컴퓨팅에서 주목받고 있다[2].

그러나 HDFS는 기본적으로 3개의 데이터 중복 저장을 통하여 가용성을 보장하는 구조를 갖고 있다. 중복 저장으로 인한 인프라 구축비용의 증가는 결국 운용비용의 증가로 이어져 서비스 경쟁력을 저하시키는 요인이 된다. 따라서 가용성을 유지하면서 데이터의 중복 저장으로 인한 저장 공간의 낭비를 최소화시키는 것은 우선적으로 해결해야 할 문제이다.

본 논문에서는 HDFS의 중복 데이터 문제를 해결하기 위해 MD5 해쉬 기술을 적용한 중복 데이터 제거 기법을 제안한다.

2. 관련 연구

2.1. Hadoop Distributed File System(HDFS)

HDFS는 분산 컴퓨팅 플랫폼인 Hadoop에서 분산 컴퓨팅 프레임워크를 지원하기 위해 개발된 분산 파일 시스템이다. HDFS는 리눅스 기반 저비용 서버를 이용한 분산 컴퓨팅에서 널리 사용된다. HDFS는 단일 네임노드와 데이터 노드로 구성되며, 이는 클라이언트 파일에 대한 접근과 데이터 저장소를 관리한다. HDFS는 64메가바이트 크기의 블록들로 파일을 관리하며, 이는 각 데이터 노드에 분산되어 저장된다. 해당 데이터노드는 각각의 블록을 로컬 파일 시스템에 파일로 관리한다. 발생 가능한 데이터노드의 장애에 대처하기 위해 블록들의 복제 본은 여러 데이터노드들에 중복되어 저장된다. HDFS는 기본적으로 관리하는 3개의 복제 본에 대하여, 한 서버 랙의 서로 다른 데이터 노드에 1개씩 저장하고, 나머지 한 개의 복제 본은 다른 랙에 따로 저장한다. 이러한 저장 관리 정책과 더불어 빠른 읽기 연산을 위해, 클라이언트의 읽기 요청 시 가장 가까운 복제 본을 선택하는 정책을 사용한다.

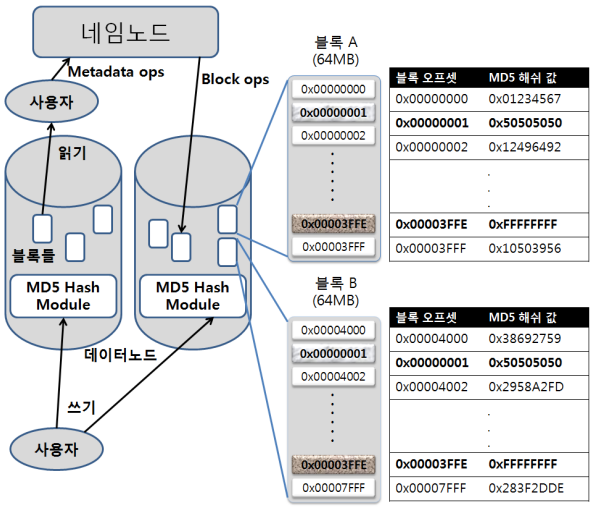
2.2. 중복 데이터 제거 기술 (De-duplication)

중복 데이터 제거 기술은 저장 공간의 효율성을 높이기 때문에 최근 주목을 받고 있다[3, 4]. 이를 위해 블록들 간의 데이터 패턴을 찾는 기술로써 MD5 해쉬 함수는 안정성 및 실용성 측면에서 뛰어나다[5]. MD5 해쉬 함수는 데이터를 512비트 단위로 나누어, 각 데이터를 해쉬 값으로 변환한다. MD5 해쉬 과정에 앞서, 데이터는 512비트의 배수가 되도록 패딩비트가 추가된다. 패딩 과정이 끝난 확

장된 데이터는 512비트 단위로 나누어져 해쉬 함수 모듈인 H_{MD5}에 입력된다. H_{MD5}는 4라운드로 구성되며, 각 라운드는 16단계로 처리된다. 최종적으로 512비트 블록이 차례로 처리되어 128비트의 해쉬 값을 출력한다.

3. 제안한 중복 데이터 제거 기법

앞에서 언급한 바와 같이 HDFS에서 사용되는 일반적인 블록 크기는 64메가바이트이다. HDFS에서는 이 블록들을 가용성을 위해 복제해서 저장하므로 저장 공간의 낭비를 초래한다. 따라서 본 논문에서는 가용성을 유지하면서 저장 공간 낭비를 최소화하기 위한 중복 데이터 제거 기법을 제안한다. 제안한 기법은 다음의 그림 1과 같다.



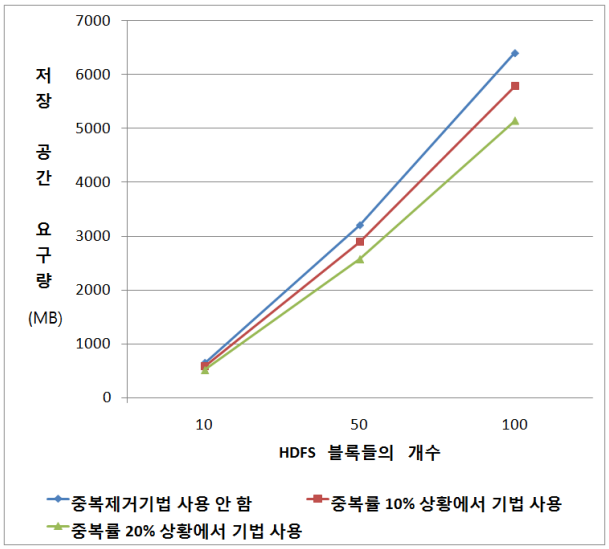
(그림 1) 중복 데이터 제거 기법을 적용한 HDFS의 구조

먼저 로컬 파일 시스템 블록 크기로 나눠진 HDFS 블록에 대해 해쉬 값을 계산한다. 계산된 블록들의 해쉬 값은 해쉬 테이블을 통해 관리된다. 로컬 파일 시스템에 존재하는 해쉬 테이블은 쓰기 연산이 수행될 때마다 새 블록들의 해쉬 값과 비교과정을 수행한다. 만약 해쉬 값이 같으면, 해당 블록들의 오프셋을 같게 해 줌으로써 중복 데이터 저장을 피할 수 있다.

본 기법을 사용 할 경우, 해쉬 테이블을 유지하기 위해 로컬 파일 시스템의 한 블록 당 128비트의 해쉬 값을 저장하기 위한 공간이 필요하다. 그러나 중복이 발생할 경우 한 블록 당 로컬 파일 시스템 블록 사이즈(4킬로바이트)만큼의 공간을 줄일 수 있다. 그 결과 효율적인 저장 공간이 사용이 가능하다.

4. 실험 결과 및 분석

본 논문에서는 검증을 위해 임의로 중복된 데이터를 만들어, 각 경우에 대해 필요한 저장 공간을 테스트하였다. 본 기법을 사용 할 경우와 아닐 경우에 대하여 저장 공간 요구량을 테스트하였다. 결과는 다음 그림 2와 같다.



(그림 2) 실험 결과

위와 같이 중복 제거 기법을 사용할 경우 중복률 10%인 상황에서는 약 9.65%, 중복률 20%인 상황에서는 약 19.69%의 저장 공간이 감소하는 것을 볼 수 있다. 실제 대용량 분산 파일 시스템에 적용할 경우, HDFS 블록들의 개수가 더 증가하므로 더 높은 성과를 기대할 수 있다.

5. 결론 및 향후 연구 방향

본 논문에서는 클라우드 컴퓨팅을 위한 분산 파일 시스템에서 저장 공간 낭비를 줄이기 위해 중복 제거 기법을 제안하였다. 실험 결과와 같이 제안한 중복 제거 기법을 사용할 경우 저장 공간 요구량을 줄일 수 있었다. 그러나 쓰기 연산이 수행될 때마다 매번 MD5 해쉬 값을 계산함으로써 쓰기 속도가 느려지는 단점이 발생한다. 이를 해결하기 위해 해쉬 함수 최적화 연구를 진행 할 것이다.

참고문헌

[1] Cloud computing, wikipedia, http://en.wikipedia.org/wiki/Cloud_computing.

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

[3] P Kulkarni, F Douglis, J LaVoie, JM Tracey, “Redundancy Elimination Within Large Collections of Files”, In Proceedings of the USENIX Annual Technical Conference, June 2004.

[4] C Policroniades, I Pratt, “Alternatives for Detecting Redundancy in Storage Systems data”, In proceedings of the USENIX Annual Technical Conference, June 2004.

[5] R. Rivest. "The MD5 Message-Digest Algorithm", RFC 1321, 1992.