

맵리듀스의 처리 속도 향상을 위한 데이터 접근 패턴에 따른 핫-데이터 복제 기법

A Hot-Data Replication Scheme Based on Data Access Patterns for Enhancing Processing Speed of MapReduce

손 인 국, 류 은 경, 박 준 호, 복 경 수, 유 재 수
충북대학교 전기·전자·정보·컴퓨터학부 정보통신
공학전공

Ingook Son, Eunkyung Ryu, Junho Park, Kyoungsoo Bok,
Jaesoo Yoo
Chungbuk National Univ.

요약

최근 대규모 데이터의 처리와 관리를 위한 분산 저장 및 처리 시스템의 연구 및 활용이 중요해지고 있다. 대표적인 분산 저장 및 처리시스템으로써 하둡(Hadoop)이 널리 활용되고 있다. 하둡 분산 파일 시스템을 기반으로 수행되는 맵-리듀스에서 테스트 할당은 데이터의 로컬리티를 고려하여 최대한 가깝게 할당한다. 하지만 맵-리듀스에서의 데이터 분석 작업에서 작업 형태에 따라 빈번하게 요청되는 데이터가 존재한다. 이러한 경우, 해당 데이터의 낮은 로컬리티로 인해 수행시간 증가 및 데이터 전송의 지연의 문제점을 야기 시킨다. 본 논문에서는 맵-리듀스의 처리 속도 향상을 위한 데이터 접근 패턴에 따른 핫-데이터 복제 기법을 제안한다. 제안하는 기법에서는 데이터 접근 패턴에 따라 높은 접근 빈도를 보이는 핫-데이터에 대한 복제본 최적화 알고리즘을 활용하여 데이터 로컬리티를 향상시키고 결과적으로 작업 수행시간을 감소시킨다. 제안하는 기법은 기존 기법에 비해 모든 노드의 데이터 이동이 감소하여 접근빈도의 분포가 균형적인 것을 확인하였다. 성능평가 결과, 기존 기법에 비해 접근 빈도의 부하가 약 8% 감소하는 것을 확인하였다.

1. 서론

최근 소셜 미디어의 성장과 모바일 장치와 같은 디지털 기기가 삶의 전반에 배치되면서 일상 속에서 다양한 종류의 대규모 데이터가 급속히 생성, 유통, 저장되고 있다. 이와 함께 방대한 데이터를 축적하고 분석하려는 노력이 이어지고 있다. 디지털로 축적된 다양한 데이터의 분석을 통해 기존에 보지 못했던 새로운 가치들이 체계적으로 도출되고 있기 때문이다. 몇 해 전부터 꾸준히 이슈화된 빅 데이터를 본격적으로 활용하기 위한 궤도에 올라서고 있는 모양새다.

이렇게 기하급수적으로 늘고 있는 데이터를 상용 솔루션으로 감당하기엔 그에 따른 비용 부담 또한 기하급수적으로 늘어날 수밖에 없다. 그러므로 대규모 데이터 처리와 관리를 위한 분산 저장 및 처리 시스템의 연구 및 활용이 중요해지고 있으며, 최근 그 해답을 오픈소스 솔루션에서 얻고 있다. 하둡(Hadoop)은 대규모 자료의 저장 및 처리를 위한 분산 응용 프로그램을 지원하는 대표적인 오픈소스 소프트웨어 프레임워크이다. 하둡은 페타바이트 이상의 대규모 데이터를 클러스터 환경에서 저장하기 위한 하둡 분산 파일 시스템(HDFS)과 이를 기반으로 병렬 처리를 지원하기 위한 맵-리듀스(MapReduce) 프레임워크로 구성된다. 하둡 분산 파일 시스템과 맵-리듀스는 실제 사례에서 확인된 것과 같이 높은 활용성으로 인해 많은 주목을 받고 있으며, 추가적인 성능 향상을

위한 연구가 활발하게 진행되고 있다.

맵-리듀스 프레임워크는 함수형 프로그래밍에서 일반적으로 활용되는 Map과 Reduce 함수 기반으로 구성된다. Map 단계에서는 함수의 정의에 따라 청크(데이터 블록)를 읽어서 가공된 데이터를 카운트의 형태로 변화하는 작업을 수행하고, Reduce 단계에서는 Map 단계의 결과를 병합하여 출력한다.

하둡 분산 파일 시스템은 데이터의 유실이나 장애 발생을 고려하여 기본적으로 3개의 청크 복제본을 노드, 랙, 오프-랙에 저장한다. 하둡 분산 파일 시스템을 기반으로 수행되는 맵-리듀스에서의 테스트 할당은 데이터의 로컬리티를 고려하여 최대한 가깝게 할당한다. 다시 말해, 테스트 수행에 필요한 청크가 위치한 노드에 테스트를 할당하거나 해당 노드가 다른 테스트를 수행 중일 경우 노드가 위치한 랙의 다른 노드로, 노드가 위치한 랙의 다른 노드도 모두 다른 테스트를 수행 중일 경우 다른 랙의 테스트를 수행 중이지 않은 노드로 할당한다. 이때, 처리 대상 데이터가 다른 노드에 위치할 경우, 처리 노드로 해당 청크를 불러오기 위한 데이터 검색, 접근, 전송 등의 추가적인 작업을 필요로 하고, 이는 해당 작업 수행 시간의 증가를 야기한다. 뿐만 아니라, 다수의 노드로 구성되는 네트워크상의 한정된 대역폭을 점유함으로써 다른 데이터의 전송 역시 지연시키는 문제가 발생한다.[1,2]

맵-리듀스에서의 데이터 분석 작업에서 작업 형태에 따라 빈번하게 요청되는 데이터가 존재한다. 예를 들어, 특정 사전 데이터와 비교를 수행할 경우, 기준이 되는 사전 데이터는 대부분의 노드에서 필요로 하게 된다. 데이터 비교를 수행할 경우 사전 데이터의 로컬리티가 낮은 상태에서 수행시간 증가 및 데이터 전송의 지연의 문제점을 야기시킨다.

이러한 문제점을 고려하여, 본 논문에서는 맵-리듀스의 처리 속도 향상을 위한 데이터 접근 패턴에 따른 핫-데이터 복제 기법을 제안한다. 제안하는 기법에서는 데이터 접근 패턴에 따라 높은 접근 빈도를 보이는 핫-데이터에 대한 복제본 최적화 알고리즘을 활용하여 데이터 로컬리티를 향상시키고 결과적으로 작업 수행시간을 감소시키는 것이 가능하다.

2. 제안하는 기법

본 절에서는 맵-리듀스의 처리 속도 향상을 위한 데이터 접근 패턴에 따른 핫-데이터 복제 기법을 제안한다. 맵-리듀스 프레임워크에서는 클라이언트로부터 제출된 작업은 잡 트래커(JobTracker)에 의해 태스크로 분할되어 태스크 트래커(TaskTracker)에 할당된다. 태스크 수행에 필요한 데이터를 찾기 위해 태스크 트래커는 하둡 분산 파일 시스템에 접근하여 메타 데이터를 활용한다. 제안하는 기법의 수행을 위해, 데이터를 관리하는 네임노드(NameNode) 상에 태스크의 데이터 접근 빈도를 파악하기 위한 모듈을 탑재하고, 표 1과 같은 접근 패턴 저장 구조를 생성한다. 표 1에서 파일 이름은 원본 파일의 위치를 포함한 이름이며, 복제본 리스트는 복제본의 위치를 포함한 이름의 리스트, 블록 아이디는 데이터 노드에서의 복제본이 저장된 블록의 위치를 나타내며 접근 수는 각 복제본에 접근된 수를 나타낸다.

표 1 . 데이터 접근 패턴 테이블

Filename	Replica_list	block_id	accessCount	storageTime
파일 이름	복제본 리스트	블록 아이디	접근 수	데이터 저장시간

접근 빈도에 따른 최적의 복제본의 수를 도출하기 위해 평균 태스크 처리 시간당 접근 빈도를 사용한다. 여기서 평균 태스크 처리 시간은 식 (1)을 사용하여 산출된다. 식(1)은 하둡 프레임워크가 클라이언트로부터 제출된 잡 처리 시간당 모든 노드에 할당된 전체 태스크를 의미한다.

$$Time_{avg} = \frac{Total\ time\ of\ Jobprocess}{Number\ of\ Total\ tasks} \quad (1)$$

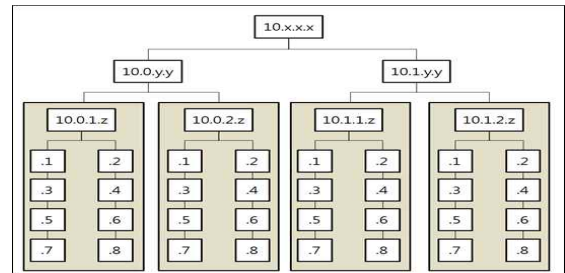
태스크 당 평균 처리 시간을 기반으로 최적의 복제본의 수를 산출하는 것이 가능하다. 식 (2)는 평균 태스크 처리 시간, 접근 수 그리고 데이터 저장 시간을 사용하여 각 데이터의 복제본의 수를 산출하는 수식이다.

$$Replica_{count} = \frac{access\ Count \times Time_{avg}}{storage\ Time} \quad (2)$$

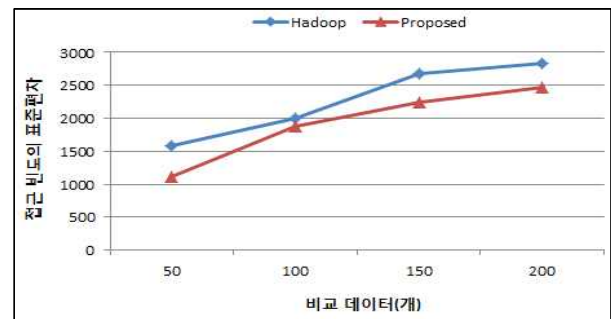
제안하는 기법에서는 최적의 복제본의 수를 산출한 후, 복제본을 임의로 배치하는 것이 아닌 로컬리티를 최대한 고려한 배치 전략을 취한다. 하둡 프레임워크는 그림 1과 같이 내부적으로 거리에 따라 규칙적으로 IP 주소를 할당한다. 구체적으로 주소 앞부분이 일치하면 할수록 근거리에 있는 노드라고 할 수 있다. 이러한 거리에 따른 IP 주소 할당을 이용하여 복제본을 배치한다. 복제본을 균등하게 배치하기 위해 복제본의 배치는 기존 데이터 배치 노드의 주소를 검사하여 데이터가 존재하지 않은 곳에 배치한다. 예를 들어, 10.0.1.1 노드에만 데이터가 존재한다면 10.1.Y.Y에 복제본을 우선 배치한다. 이러한 복제본 배치 정책으로 데이터의 로컬리티를 최대한 할 수 있다[3].

3. 성능 평가 및 분석

본 절에서는 제안하는 기법의 우수성을 입증하기 위해, 하둡과의 시뮬레이션을 통한 성능 평가를 수행하였다. 본 시뮬레이션은 클러스터 내의 전체 노드 수 100개, 전체 청크 데이터를 1000개로 고정된 후, 태스크 수행에 필요한 비교 데이터를 50~200개로 변경하며 전체 노드의 접근 빈도의 표준편차를 비교 평가하였다. 표준편차는 접근빈도의 분포를 의미한다. 그림 2에서 기존기법과 비교하여 제안하는 기법이 표준편차가 전체적으로 감소한 것을 확인 할 수 있다. 성능평가 결과, 기존기법에 비해 제안하는 기법에서 접근 빈도의 부하가 약 8% 감소하는 것을 확인하였다.



▶▶ 그림 1. 거리에 따른 주소 체계



▶▶ 그림 2 . 비교 데이터 수에 따른 접근 빈도 표준편차

4. 결론 및 향후 연구

본 논문에서는 기존 기법의 문제점을 분석하고 맵-리듀스의 처리 속도 향상을 위한 데이터 접근 패턴에 따른 핫-데이터 복제 기법을 제안하였다. 제안하는 기법에서는 데이터 접근 패턴에 따라 높은 접근 빈도를 보이는 핫-데이터에 대한 복제본 최적화 알고리즘을 활용하여 데이터 로컬리티를 향상시키고 작업 수행시간을 감소시켰다. 성능평가 결과, 제안하는 기법은 기존 기법에 비해 접근 빈도의 부하가 약 8% 감소하는 것을 확인하였다. 향후 연구로는 실제 하둡 환경에서의 데이터 이동과 접근 빈도의 성능 평가를 수행하는 것이다.

■ 참고 문헌 ■

[1] J. Dean and S. Ghemawat "MapReduce: Simplified Data Processing on Large Cluster," Communications of the ACM Vol.51, No. 1, pp.107-113, 2008.
 [2] Hadoop, <http://hadoop.apache.org>
 [3] The Hadoop Distributed File System: Architecture and Design, http://hadoop.apache.org/core/docs/current/hdfs_design.html