대용량 스트리밍 데이터에 대한 분산 인메모리 기반의 점진적 처리 기법

In-memory based Incremental Processing Method for Massive Streaming Data

육미선, 김병훈, 한지은, 노연우, 임종태, 복경수, 유재수
충북대학교 정보통신공학과

Misun Yook, Byounghoon Kim, Jieun Han, Yeonwoo Noh, Jongtae Lim, Kyoungsoo Bok, Jaesoo Yoo
Chungbuk National University

요약

본 논문에서는 스트리밍 데이터에 대한 점진적 연산을 지원하는 실시간 분산 인메모리 데이터 처리 기법을 제안한다. 제안하는 데이터 처리 기법은 기존에 처리된 데이터를 인메모리에 유지하고 새로운 스트리밍 데이터가 입력되었을 때 기존에 처리된 데이터를 재사용한다. 성능평가를 통해 제안하는 기법이 하둡에 비해 대용량 스트리밍 데이터를 빠르게 처리할 수 있음을 보인다.

I. 서론

최근 데이터가 기하급수적으로 증가함에 따라 대용량 데이터를 처리에 관한 연구가 많이 수행되고 있다. 전 세계에서 트위터는 2014년 기준 하루 약 5억개의 트윗이 발생하고 페이스북은 하루 약 25억개 이상 게시되고 있다. 빅데이터의 발달로 하둡과 같은 빅데이터를 처리하기 위한 프레임워크가 등장하였다. 하둡의 경우 일괄 배치처리시스템으로 대용량 데이터를 효율적으로 처리할 수 있다. 따라서 하둡을 이용하여 대용량 데이터 처리를 하게 되었다.

스트리밍 데이터가 계속적으로 생성되면서 실시간 처리가 필요하게 되었다. 스트리밍 데이터 처리는 슬라이딩 윈도우를 이용한 연산을 한다. 이러한 슬라이딩 윈도우 처리과정은 타임 윈도우가 겹쳐서 처리되기 때문에 기존 데이터를 재사용할 필요가 있다. 하지만 하둡은 스트리밍 데이터 처리를 지원하지 않는다. 따라서 이러한 처리를 위한 다양한 연구가 진행되고 있다[1]. 하지만 [1]은 데이터를 재사용하지 않고 처리하여 속도의 향상을 기대하기 어렵다. [2]에서는 인메모리 기반 재계산 기법을 제안하였지만 스트리밍 데이터 처리하지 못한다. 대표적인 스트리밍 데이터 처리 프레임워크로는 스톰[3]과 스파크[4]를 들 수 있다. 스톰은 스트리밍 처리를 지원하지만 과거 처리된 데이터를 재사용하지 못한다. 스파크는 일괄 처리 방식으로 반복 작업을 처리하는데 목적이 있어 슬라이딩 윈도우를 처리하지 못한다.

본 논문에서는 실시간으로 들어오는 스트리밍 데이터에 대한 점진적 처리를 제공하기 위한 기법을 제안한다. 제안하는 기법은 인메모리 기반 처리 기법으로 기존에 처리된 데이터를 재사용한다.

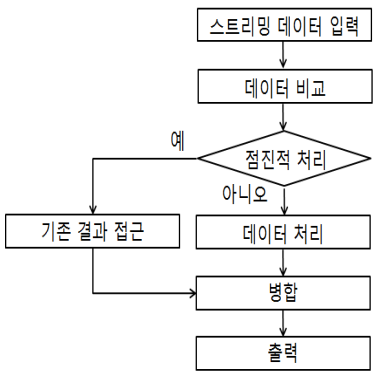
* 교신저자 : yjs@chungbuk.ac.kr

이 논문은 2013년도 정부(미래창조과학부)의 재원으로 한국연구재단의 지원(No.2013R1A2A2A01015710)과 미래창조과학부 및 정보통신기술진흥센터의 대학ICT연구센터육성 지원사업(IITP-2015-H8501-15-1013), 한국과학기술정보연구원 "초고성능컴퓨팅 기반 건강한 고령사회 대응 빅데이터 기술 개발 (K-15-L03-C02)" 연구사업의 연구비 지원으로 수행되었음

Ⅱ. 제안하는 기법

1. 전체 시스템 구조

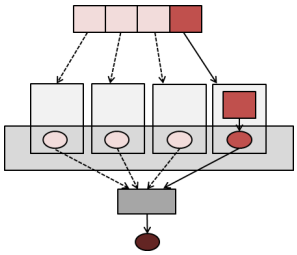
본 논문에서는 대용량 스트리밍 데이터 처리 기법을 제안한다. 제안하는 기법은 기존에 처리된 데이터를 인메모리에 저장하고 재사용하여 데이터의 처리 속도를 향상시킨다. 디스크 저장 대신 인메모리를 사용하여 디스크 I/O 발생을 감소시켜 실시간으로 데이터를 빠르게 처리할 수 있다. 그림1은 제안하는 스트리밍 데이터 처리 과정을 나타낸다. 스트리밍 데이터가 입력되면 데이터를 비교하여 기존에 처리된 데이터가 있으면 기존 결과에 접근하여 데이터 처리 결과를 가져온다. 가져온 데이터는 새로 처리된 데이터와 병합하여 결과를 출력한다.



▶▶ 그림 1. 제안하는 기법의 스트리밍 데이터 처리 과정

2. 점진적 처리 과정

제안하는 처리 기법은 데이터 재사용을 통해 빠른 처리 성능을 제공한다. 그림2는 점진적 처리 과정을 나타낸 것이다. 점진적인 처리 과정은 데이터가 입력되면 데이터 비교를 통해 같은 데이터를 찾는다. 같은 데이터가 있으면 그 처리 결과가 있는 인메모리에서 처리 결과를 가져오고 같은 데이터가 없으면 데이터를 처리한다.

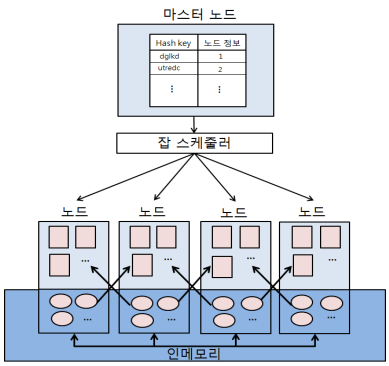


▶▶ 그림 2. 점진적 처리 과정

3. 잡 할당

처음 스트리밍 데이터가 입력되면 데이터 비교를 위한 해시 키 값을 생성한다. 해시 키 값 생성 이후 데이터는 잡 스케줄러에 의해 분산 노드에 할당되는데 잡 스케줄러는 라운드 로빈 스케줄링 기법을 이용한다. 노드에서 처리된 데이터는 각 노드의 인메모리에 유지하고 결과를 병합하여 출력한다. 각 노드마다 메모리 영역을 할당하여 데이터 처리 결과를 유지하고, 마스터 노드에 입력 데이터 정보인 해시 키 값과 데이터를 처리한 노드 정보를 유지한다.

계속적으로 스트리밍 데이터가 입력되면 기존 스트리밍 데이터의 처리와 마찬가지로 데이터 비교를 위한 해시 키 값을 생성한다. 생성된 해시 키 값은 마스터 노드에 저장된 기존 해시 키 값과 비교하여 같은 값을 찾는다. 그림 3은 데이터 저장 기법을 나타낸다. 노드에서 처리된 데이터는 인메모리에 저장되고 저장된 데이터는 마스터 노드에서 관리한다. 마스터 노드에 저장된 기존 해시 키 값과 비교하여 동일한 해시 키 값이 존재하면 해당 노드에서 데이터를 가져오고, 기존에 처리된 데이터가 없으면 새로 노드를 할당하여 데이터를 처리한다.

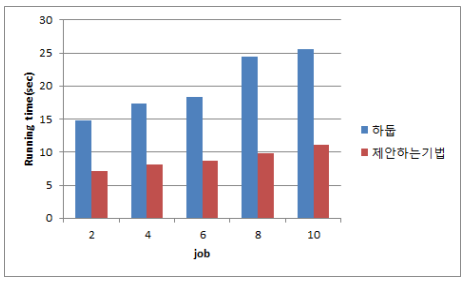


▶▶ 그림3. 데이터 저장 기법

Ⅲ. 성능평가

본 논문에서는 제안하는 기법의 우수성을 보이기 위해 하둡과의 성능 평가를 수행하였다. 임의의 데이터를 생성하여 데이터의 개수에 따른 성능 향상을 비교하였다. 성능 평가는 속도를 평가하였다. 성능평가 환경은 Intel® CoreTM i3 CPU 2100 프로세서 4G 메모리, 500GB 디스크 환경에서 4개의 가상 노드로 구성된 분산 환경으로 구성된다.

그림 4는 잡 수에 따른 수행시간을 비교한 결과이다. 성능 평가 결과 제안하는 기법이 하둡에서 처리보다 평균 약 55% 속도 향상을 보였다. 제안하는 기법에서는 기존에 사용된 데이터를 재사용함으로써 수행시간을 단축할 수 있었다.



▶▶ 그림 4. 잡 수에 따른 수행시간 비교

Ⅳ. 결론

본 논문에서는 인메모리 기반 데이터의 점진적인 처리 기법을 제안하였다. 제안하는 기법은 과거에 처리되었던 데이터를 메모리에 저장하고, 실시간으로 스트리밍 데이터가 입력되었을 때 해시 키 값으로 비교하여 이미 처리된 데이터를 활용하여 처리를 한다. 그러므로 기존 기법에 비해 빠른 데이터 처리 속도를 보인다. 성능평가를 통해 제안하는 기법이 하둡에 비해 평균 약 55% 향상됨을 확인할 수 있었다. 향후 연구로는 제안하는 기법을 실제 스트리밍 데이터 처리에 적용할 예정이다.

▌참 고 문 헌▐

[1] F. Zhang, J. Cao, S. U. Khan, K. Li, and K. Hwang, “A task-level adaptive MapReduce framework for real-time streaming data in healthcare applications,” Future Generation Computer Systems, vol.43-44, pp.149-160, 2015

[2] D. Tiwari and Y. Solihin, “MapReuse: Reusing Computation in an In-Memory MapReduce System,” Proc. International Parallel and Distributed Processing Symposium, pp.61-71, 2014

[3] https://storm.apache.org/

[4] https://spark.apache.org/