

분산 시스템에서 점진적 갱신을 사용한 향상된 Prefetching

송경태, 김재형, 박상현

연세대학교 컴퓨터과학과

e-mail : skt8776@yonsei.ac.kr, { jaehyungkim, sanghyun } @ cs.yonsei.ac.kr

Enhanced Intra-block Prefetching using incremental update in Distributed System

Kyungtae Song, Jaehyung Kim, Sanghyun Park

Yonsei University, Computer Science Department

요 약

구글이 발표한 구글 파일시스템과 맵리듀스 논문 이후로 여러 오픈소스 분산시스템들이 나타남에 따라 상용시스템을 이용하지 않아도 대용량의 데이터를 처리 할 수 있는 서비스들이 등장하고 있다. 특히 분산 환경에서 병렬 처리를 제공하는 맵리듀스 프레임워크를 사용한 데이터 처리 방식은 가장 보편적인 개발 방법이다. 하지만 기존 상용 시스템에 비하여 처리 속도가 느린 단점이 있어 이를 보안하기 위해 학계 및 업계에서 많은 시도가 있어 왔다. 특히 데이터 지역성(data locality)를 향상시키기 위해 많은 시도들이 이루어져 왔고, 그 중 대표적인 것이 Prefetching 기법이다. 기존 Prefetching 기법에서는 Data의 Computation 시점과 정 반대의 시점에서 진행되므로, Prefetching이 완료되기 전까지 Prefetching의 이득을 얻을 수 없다는 단점이 있다. 그래서 본 논문에서는 Prefetching 방법을 개선하여 수행속도와 Prefetching 속도에 따른 동적 prefetching 기법을 소개하고자 한다.

1. 서 론

가트너(Gartner)가 빅데이터 용어를 3V(Volume, Velocity, Variety)로 정의한 이후 빅데이터에 대한 관심이 뜨겁다. 특히 아파치(Apache) 재단[1]을 중심으로 다양한 오픈소스 분산 처리 시스템이 대중에게 공개되면서 많은 이들이 저렴한 비용으로 빅데이터를 처리할 수 있는 환경을 갖출 수 있게 되었다. 이로 인해 많은 연구자들 및 개발자들이 빅데이터 오픈소스 관련 연구, 개발을 하고 있다. 하둡(Hadoop)[2]은 대표적인 오픈소스 빅데이터 플랫폼으로서, 분산 처리시스템 MapReduce framework와 분산 파일 시스템 HDFS(Hadoop Distributed File System)로 구성된다. MapReduce는 수행해야 할 작업을 여러 개의 작은 단위로 쪼개어 서로 다른 서버에서 병렬적으로 연산을 할 수 있도록 해주는 프로그래밍 모델이다.

MapReduce의 작동방식은 다음과 같다. HDFS에 처리해야 할 데이터가 저장되어 있고, 그것을 나누어 놓은 것을 input split이라 부른다. Input split은 블록의 크기로 제한되고, 서버에서 수행되는 각 작업들 중 Map 작업을 수행하는 노드는 하나의 input split에 할당되어

연산을 수행하게 된다. Map 작업으로 생성된 결과물을 중간 결과물(Intermediate output)이라고 부르고, 중간 결과물들은 combine, shuffle의 작업을 거쳐 reduce 작업을 수행하는 노드로 전달된다.

데이터 Prefetching이란, 데이터 locality를 활용하기 위하여 연산에 쓰이고 있지 않은 데이터이더라도 추후에 쓰일 가능성이 있는 데이터를 미리 캐시 등으로 옮겨 놓는 것을 말한다. 데이터 locality를 활용하기 위한 다양한 시도들이 꾸준히 있었기 때문에 기존 연구들은 쉽게 접할 수 있다. [2][3][4] 그 중 HPMR - Seo et al.[5]의 논문은 MapReduce Framework 상에서 Prefetching과 Preshuffling을 이용하여 MapReduce 연산 속도를 향상시킨 논문이다. 특히 논문에서 제시된 intra-block prefetching 방법은 Bi-directional prefetching으로써, prefetching으로 인한 부작용이 생기지 않는 방법이다. 하지만, 본 연구자는 부작용을 생각해서 만든 방법 때문에 prefetching으로 취할 수 있는 이득이 적어졌다는 것을 발견하게 되었다.

이에 본 연구는 prefetching의 부작용이 발생하지 않으면서 prefetching으로 얻을 수 있는 이득을 최대화하는 방법에 대하여 다뤄보자 한다. 수식을 통해 prefetching을 시작할 수 있는 적당한 지점을 계산하고, 그 계산된 지점부터 bi-directional prefetching이 아닌 one directional prefetching을 이용하여 최대한 많은 양의

※ 이 논문은 2015년도 정부(미래창조과학부)의 재원으로 한국연구재단의 지원을 받아 수행된 연구임 (NRF-2015R1A2A1A05001845).