MapReduce를 이용한 Hadoop기반의 TPC 오프라인 트래킹 시스템 제안

이수강, 윤성호, 심규석, 김성민, 김명섭
고려대학교
{sukanglee, sungho_yoon, kusuk007, gogumiking, tmskim}@korea.ac.kr

Proposal of Hadoop-based MapReduce TPC Offline Tracking System

Lee Su-Kang, Yoon Sung-Ho, Shim Kyu-Seok, Kim Sung-Min, Kim Myung-Sup
Korea Univ.

요 약

대형 강입자 충돌기(Large Hardron Collider)는 우주 빅뱅 직후의 고 에너지 상태 재현을 위해 빛의 속도에 가깝게 양성자를 가속해 충돌시키는 장치로서 전 세계 80여 개국, 8000여명의 세계 물리학자들이 참여하고 있는 세계 최대의 고에너지 물리 실험 장치이다. LHC에서 실행된 ALICE 실험은 중이온입자 충돌 실험이며 TPC(Time Projection Chamber)는 ALICE 실험에서 사용된 Main Detector중 하나이다. TPC 에서는 입자 충돌 실험시 수백만의 입자데이터가 생성되며 저장된다. 저장된 입자데이터는 TPC Offline Tracker에 의해 재구축 된다. 본 논문에서는 TPC Offline Tracker의 알고리즘을 과정별로 설명하고 이를 적용한 Hadoop기반의 MapReduce TPC Offline Tracker 시스템을 구조를 제안한다. 그리고 제안된 시스템의 한계점으로 밝혀진 Mapper 노드와 Shared Storage 서버 사이의 네트워크 오버헤드 문제를 개선시킬 수 있는 향후 연구에 대해 언급하였다.

Ⅰ. 서 론

대형 강입자 충돌기(Large Hadron Collider, 이하 LHC)는 우주 빅뱅 직후의 고 에너지 상태 재현을 위해 빛의 속도에 가깝게 양성자를 가속해 충돌시키는 장치로서 전 세계 80여 개국, 8000여명의 세계 물리학자들이 참여하고 있는 세계 최대의 고에너지 물리 실험 장치이다. 그 중 ALICE 실험은 LHC에서 수행된 중이온입자 충돌실험이며 TPC(Time Projection Chamber, 이하 TPC)는 ALICE 실험에서 사용된 Main Detector중 하나이다.

TPC[1]는 지름 5미터, 길이 5미터인 실린더 형태의 용기이며 용기 내부에는 한 개의 격벽에 의해 두 공간으로 나누어져 있다. 나누어진 공간은 각각 18개의 섹터로 나누어져 있으며, 각 섹터 내부에는 이온화되기 쉬운 가스로 채워져 있다. TPC 내부에서 충돌 후 발생한 입자들은 가스를 이루는 입자를 지나가게 되고, 가스를 이루는 입자는 전자를 사방으로 방출한다. 이 전자가 사방으로 방출되어 센서에 기록이 되고, 전자가 도달한 시간을 분석하면 용기 내의 입자의 위치를 알 수 있다.

LHC에서는 한번의 실험에 약 1000억 개의 양성자를 하나의 덩어리로 만들어 가속하는데 이를 양성자 빔 번치(bunch)라고 하며 번치 와 또 다른 번치를 빛의 속도와 가깝게 가속하여 충돌시킨다. TPC에서 발생하는 데이터는 입자 한 개가 충돌할 때 약 20MB의 데이터가 발생한다. TPC에서는 충돌 후 수백만의 입자가 생성되며 이러한 입자의 위치를 찾아내는 것은 고도화된 분석법이 필요하다. 이를 위해 슈퍼 컴퓨터 또는 그리드 컴퓨팅(Grid Computing) 환경에서 분석을 하고 있으며 여러 개의 GPU를 이용하여 병렬로 입자의 위치를 찾고 추적하는 연구도 활발히 진행 중이다.

본 논문은 TPC에서 발생하는 원시데이터를 Hadoop기반의 MapReduce 연산을 통해 Offline 환경에서 TPC 트래킹 시스템을 제안하고 제안된 시스템의 한계점을 기술한다. 2장에서는 TPC Tracker 관련 연구들에 대해 설명하고 3장에서는 제안하는 시스템에 소개하고 한계점을 언급한다. 마지막으로 4장에서는 결론과 향후 연구를 언급한다.

Ⅱ. 관련 연구

본 장에서는 TPC에서 발생하는 데이터를 이용하는 TPC Online, Offline Tracker와 관련된 연구 및 현재 진행 되고 있는 연구에 대해 언급한다.

TPC Tracker는 Offline Tracker과 Online Tracker로 나누어지며 알고리즘 코드는 각각 독립적이지만 사용되는 입력데이터는 같다. TPC Online Tracker는 트리거링 및 온라인 분석을 위해 실험이 진행되는 동안 동작한다. 반면에 TPC Offline Tracker는 Online Tracker보다 느리지만 더욱 정확하고 많은 데이터 분석을 위해 사용된다.

관련연구[2]은 TPC에서 얻어진 Lead-to-Lead 충돌 실험의 데이터를 GPU를 사용하여 Online Tracking 작업을 수행했을 때 CPU를 사용했을 때보다 Tracking을 완료하는 시간이 평균 3배 가량 빠른 것을 보였다. 이는 TPC가 부분적으로는 병렬 처리에 적합한 디자인으로 설계되어 CPU를 이용한 Tracking보다 GPU를 이용한 병렬 처리가 가능함을 보인다. 이후 연구[3] 에서는 CPU가 처리하기 적합한 Step과 GPU가 처리하기 적합한 Step을 각각 나누어 처리함으로써 더욱더 성능이 향상된 TPC Online Tracker를 제안하였고 현재는 GPU 최적화 작업을 통해 성능 향상 연구를 하고 있다.

현재 CERN에서는 Aliroot를 이용하여 TPC Offline Tracking 분석을 하고 있으며 현재 ZeroMQ 패키지를 이용한 새로운 프로토 타입의 TPC Offline Tracker를 연구, 개발중에 있다.

Ⅲ. 제안하는 시스템 및 한계점

본 장에서는 Hadoop 기반의 MapReduce를 이용하여 TPC Offline Tracker 시스템을 제안하고, 본 논문에서 제안하는 시스템의 한계점을 언급한 후 향후 연구의 방향을 제시하고자 한다.

Hadoop은 여러 개의 컴퓨터(노드)를 하나인 것처럼 묶어 대용량의 데이터를 분산 처리하는 기술이며, 구글에서 제안한 MapReduce를 Java로 구

현한 오픈 소스 프레임워크 이다. 하둡 파일 시스템(HDFS)은 여러 개의 컴퓨터에 대용량 파일들을 나눠서 저장을 하기 때문에 데이터를 중복 저장하여 데이터의 안정성을 얻을 수 있고, 시스템 확장에 용이하다. 또한 기존의 값비싼 고성능 서버에 비해 유지보수 비용이 저렴하여 여러 빅데이터 분야에서 가장 많이 쓰이고 있다.

Hadoop 기반의 TPC Offline Tracker를 구현, 적용하여 얻고자 하는 장점은 병렬 처리 가능한 TPC Cluster Event Data를 여러 개의 노드에서 분산 병렬 처리하여 처리속도를 높일 수 있다. 그림 1은 본 논문에서 제안하는 Hadoop 기반의 TPC Offline Tracker 시스템의 구성도 이다.

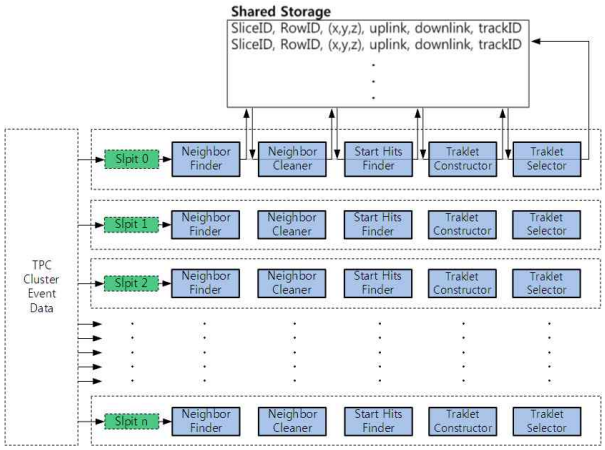

그림 1. Hadoop 기반의 TPC Offline Tracker 시스템 구성

TPC Offline Tracker 시스템은 총 5개의 Step으로 구성된다. 이 시스템에서 Reduce는 사용하지 않고 서로 다른 역할을 하는 5개의 Mapper를 ChainMapper로 연결하여 연속적으로 수행하고, 각각의 Mapper에서 수행한 작업들은 Shared Storage에 업데이트한다. 최종적으로 Shared Storage에는 각 입자(Cluster)들로 이루어진 완성된 Track 데이터가 남게 된다.

첫 번째 Step의 Mapper는 Neighbor Finder를 수행한다. Neighbor Finder에서는 Split된 TPC Cluster Event Data를 입력 데이터로 사용하게 되며 Mapper는 하나의 Cluster를 처리하게 된다. Neighbor Finder는 Mapper에서 처리되는 입자의 좌표와 주변 입자들의 좌표를 기준으로 제일 가까운 입자를 찾아 이웃관계를 Shared Storage의 UpLink 또는 DownLink 정보에 Update하게 된다.

두 번째 Step의 Mapper는 Neighbor Cleaner를 수행한다. Neighbor Cleaner는 Neighbor Finder에서 Shared Storage에 Update한 Cluster Data를 입력 데이터로 사용한다. Mapper에서 처리되는 입자와 이웃관계에 있는 입자를 비교하여 서로 이웃관계에 있지 않은 Cluster의 경우 Shared Storage의 UpLink 또는 DownLink의 정보를 삭제하여 단방향 이웃관계를 끊게 된다.

세 번째 Step의 Mapper는 Start Hits Finder를 수행한다. Start Hits Finder는 3개 이상의 Cluster가 Neighbor 하나의 이웃 관계로 연속되어 연결된 시드(seed) Track을 찾는다. 시드 Track을 찾게 되면 Start Hits Finder는 해당 Track에 TrackID를 부여하고 Shared Storage를 업데이트 한다.

네 번째 Step의 Mapper는 Tracklet Constructor를 수행한다. Tracklet Constructor는 세 번째 Step의 출력인 시드 Track을 시작으로 시드를 구성하고 있는 Cluster의 좌표를 이용하여 다음 Cluster를 찾아 시드 Track의 Member Cluster로 만든다.

다섯 번째 Step의 Mapper는 Tracklet Selector를 수행한다. Tracklet Selector는 네 번째 Step의 Tracklet Constructor의 결과를 바탕으로 Track을 최종적으로 선택한다.

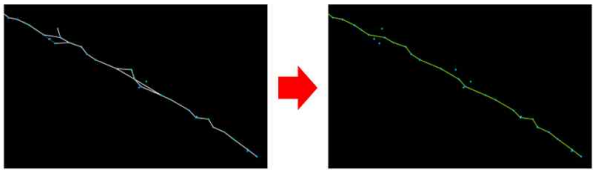

그림 2. Tracklet Selector에 의해 최종 선택된 Track

그림 2의 좌측 그림은 Trackelt Constructor까지 완료된 Track 이며 Main Track 주변에 붙어 있는 Branch Track들이 존재한다. Tracklet Selector는 Track의 길이를 바탕으로 긴 쪽의 Main Track을 유지하며 주변의 Branch Track을 정리하면서 하나의 Track을 완성시킨다. 그림 2의 우측 그림은 다섯 개의 Step을 모두 마친 완성된 Track이며 완성된 트랙에 대한 정보는 Shared Storage에 저장되고 TPC Offline Tracking은 종료된다.

하지만 본 논문에서 제안한 시스템의 Shared Storage는 구현되지 않았다. Hadoop에서 각 노드의 Mapper들은 서로 데이터를 공유할 수 없고 독립적으로 동작하기 때문이다. 본 논문에서 제안하는 시스템을 만들기 위해 Socket을 이용하여 각 Mapper들의 출력을 저장하고 이를 얻어오는 Shared Storage 서버를 구축하였지만 그에 따른 Network의 오버헤드로 인해 전체 Processing Time이 지나치게 증가하는 결과가 나타났다. 이러한 이유 때문에 CERN에서는 TPC Offline Tracker의 분산 처리를 위해 별도의 메시징 서비스인 ZeroMQ를 이용하는 연구를 하고 있는 중이다.

Ⅳ. 결론 및 향후 연구

본 논문에서는 Hadoop기반의 TPC Offline Tracker 시스템을 제안하고 이의 한계점을 언급하였다. 논문에서 제안하는 시스템을 만들기 위해 Socket을 이용하여 Shared Storage 서버를 구축하였으나 Network의 오버헤드가 전체 Processing Time을 지나치게 증가시키는 결과를 초래했다. 이러한 이유로 CERN에서는 TPC Offline Tracker를 기존의 AliRoot 기반에 새로운 메시징 서비스인 ZeroMQ를 이용한 새로운 분산 처리 연구를 진행 중이다. 따라서 본 연구진은 향후 연구로 메시징 서비스인 ZeroMQ와 본 논문에서 제안한 Hadoop기반의 TPC Offline Tracker를 접목시키는 연구를 계속 진행할 계획이다.

ACKNOWLEDGMENT

본 논문은 교육과학기술부의 재원으로 한국연구재단의 지원을 받아 수행된 BK21 플러스 사업의 연구 결과임 (No. T1300572)

참 고 문 헌

[1] Musa, Luciano. "The time projection chamber for the ALICE experiment." Nuclear Physics A 715 (2003): 843c-848c.

[2] Rohr, David, et al. "ALICE HLT TPC Tracking of Pb-Pb Events on GPUs." Journal of Physics: Conference Series. Vol. 396. No. 1. IOP Publishing, 2012.

[3] Rohr, David. On development, feasibility, and limits of highly efficient CPU and GPU programs in several fields. Diss. Frankfurt am Main, Johann Wolfgang Goethe-Univ., Diss., 2014, 2014.