빅데이터 플랫폼의 병렬성 측면에서의 HDFS Archival Storage 성능 분석

김재형 ○ 박상현
연세대학교 컴퓨터과학과
jaehyungkim@yonsei.ac.kr, sanghyun@yonsei.ac.kr

HDFS Archival Storage Performance Evaluation focused on Parallelism in Bigdata Platforms

Jaehyung Kim○ Sanghyun Park
Dept. of Computer Science, Yonsei University

요 약

본 논문에서는 대표적인 빅데이터 플랫폼인 하둡(Hadoop)에서 제공하는 Hybrid array 기술인 Archival storage를 분석하고, 빅데이터 플랫폼의 병렬성을 극대화하기 위해, Hybrid array에서의 디스크 별 대역폭을 고려한 노드(Node) 내 저장장치들(Storages)에 대한 데이터 분산 정책의 필요성을 확인한다. 이를 위해, SQL 질의 처리를 위한 빅데이터 플랫폼 중 하나인 SparkSQL에서 대표적인 분석 벤치마크 테스트인 TPC-H를 수행하여 저장장치 관리 정책에 따른 성능 차이를 확인하였으며, 실험 결과로부터 성능 차이의 발생 원인을 저장장치 관점에서 검증하고 병렬성 극대화를 위한 요인을 도출한다.

1. 서론

대용량 데이터를 처리하기 위한 빅데이터 플랫폼은 데이터 병렬성(Data Parallelism)과 태스크 병렬성(Task Parallelism)[1] 모두에 기반하고 있다. 특히, SSD(Solid State Disk)의 등장으로 I/O 처리와 CPU 연산의 성능 차이가 줄어들고, SSD의 내부 병렬성을 통해 빅데이터 플랫폼의 태스크 병렬성을 극대화할 수 있었으나, SSD의 제한된 수명 문제로 인해 HDD를 함께 사용하는 Hybrid array 기술이 발전하고 있다[2]. 대부분의 빅데이터 플랫폼이 분산 파일 시스템으로 선택하는 하둡(Hadoop)[3]의 HDFS(Hadoop Distributed File System)은 Archival storage라는 Hybrid array 기술을 공개했다. 본 논문에서는 Archival storage의 병렬성을 저해하는 데이터 관리 정책의 문제점을 대표적인 빅데이터 질의 처리 플랫폼인 SparkSQL[4]을 이용한 실험을 통해 검증하고, 빅데이터 플랫폼의 병렬성을 극대화시키기 위해 각 디스크의 대역폭을 고려한 데이터 분배 정책을 통해 Hybrid array의 전체 대역폭을 활용해야 한다는 사실을 도출하였다.

2. Archival Storage

Archival storage는 HDFS의 블록에 대한 replica를 관리하는 방식에 따라 Hot, Cold, Warm, All_SSD, One_SSD, Lazy_Persist로 나뉜다[5]. 앞의 3가지는 Archiving을 위한 정책이며, 마지막 정책은 Ramdisk를 활용하기 위한 정책으로 고려대상에서 제외된다. All_SSD 정책 역시 SSD에 남은 용량이 없을 경우에만 HDD로 전송하므로 제외된다. 본 논문에서 사용하는 One_SSD 정책은 replica 중 1개는 SSD에 저장하고, 남은 n-1개의 replica를 HDD에 저장하는 방식으로 전형적인 Hybrid array 기술에 해당한다.

3. Archival storage 실험 분석

본 섹션에서는 저장장치 정책에 따른 질의 처리 성능의 차이를 확인하기 위한 벤치마크 테스트 결과를 제시하고, 질의에 포함된 테이블의 실제 데이터가 어떤 방식으로 다수의 저장장치에 분배되어있고, 또 질의 처리 과정에서 각 데이터에 대한 접근 방식을 검증한다.

3.1. 실험환경

본 실험에서는 총 4대의 서버를 활용해 클러스터를 구성하였으며, 각 서버의 하드웨어 환경은 아래 표 2와 같다. 각각의 서버는 10 Gigabit 네트워크 환경에서 연결되었다.

표 1. 하드웨어 환경

CPU	14 physical core (Hyper-threading 14 logical core) × 2 ea
Memory	128 GB
SSD	550 MB/s × 2 ea 550 MB/s × 1 ea (중간질의결과 저장)
HDD	110 MB/s × 10 ea
Network	10 Gigabit Ethernet

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

또한, 1대의 서버는 HDFS 클러스터 구성을 위해 NameNode 전용으로 사용되었으며, 나머지 3대의 서버에 DataNode가 설치되었다.

SparkSQL은 YARN(Yet Another Resource Negotiator)를 통해 동작하며, HDFS의 NameNode와 같은 서버에 Resource Manager를 할당하였으며, 나머지 3대의 서버에 Node Manager를 할당하였다. Archival storage 적용을 위해 2.7.2 버전 기준 하둡을 사용하였으며, SparkSQL은 1.6.1 버전을 사용하였다. HDFS 데이터 블록의 크기는 기본값인 128MB로 설정하였으며, 블록의 Replication factor는 3이다.

3.2. TPC-H 벤치마크 결과

TPC-H 벤치마크 테스트[6]는 비즈니스 환경에서 사용되는 대용량 데이터 분석을 위한 유형의 질의의 집합으로 총 22개의 질의로 구성되어 있다. 이 중에서 저장장치의 성능을 평가하기 위해 CPU 사용률이 비교적 낮은 질의 6번을 평가 대상으로 선정하였다. 또한, 저장장치에 대한 접근을 높이기 위해 테이블을 구성하는 데이터는 인덱스를 생성하지 않는 raw 데이터 포맷을 사용하였다.

질의 6번은 4개의 필터형 조건자(Predicate)와 하나의 프로젝션(Projection)으로 구성되어 있으며, 복잡한 조인(Join) 연산이나, 집합(Aggregation) 연산자를 사용하지 않으므로 CPU 사용률이 TPC-H 벤치마크의 다른 질의에 비해 상대적으로 낮다. 또한, 질의 처리 중 발생하는 중간 질의 처리 결과(Intermediate Results)를 처리하기 위한 저장소는 모든 정책에 대해 별도 SSD를 할당하여 실험의 형평성을 고려하였다.

아래 그림 1은 질의 6번을 1) SSD ONLY: SSD 2개를 활용할 경우, 2) HDD ONLY: HDD 10개를 이용할 경우, 3) Archival Storage: 2개와 SSD와 10개의 HDD에 Archival storage의 One_SSD 정책을 적용한 경우의 수행 시간 차이를 나타낸다.

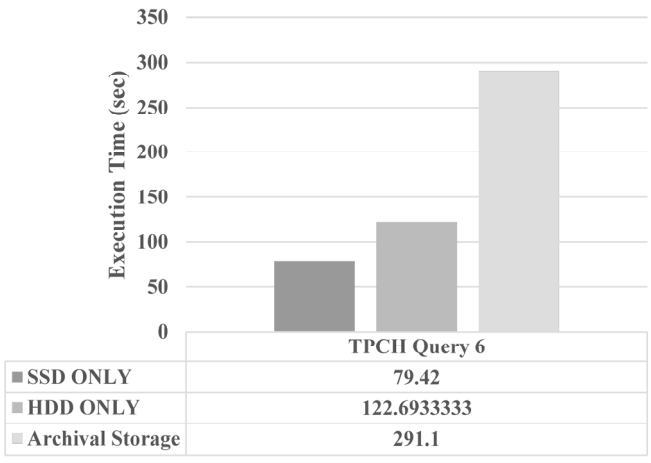

그림 1. 저장장치 정책 별 질의 6번 수행 시간

SSD 2개의 합계 대역폭은 1100 MB/s이며, HDD 10개의 합계 대역폭과 동일하다. 그럼에도 불구하고, 질의 수행 시간은 HDD만 사용했을 경우가 약 64% 더 느린 결과를 보였다. 이는 노드 내 태스크 병렬성의 결과로, 다수의 task, 즉 다수의 process들에서 발생하는 I/O 요청들이 서로 간에 간섭을 일으켜 랜덤 I/O 패턴을 보이기 때문인데, SSD 특성상 HDD에 비해 랜덤 I/O에 대한 처리가 용이하다는 측면에서 성능의 차이가 발생함을 확인할 수 있다[7].

흥미롭게도, Archival storage의 One_SSD 정책이 매우 느린 성능(SSD ONLY 대비 약 273% 느림)을 보였는데, 이러한 원인은 아래 그림 2의 대역폭 변화량을 통해 명확하게 확인할 수 있다.

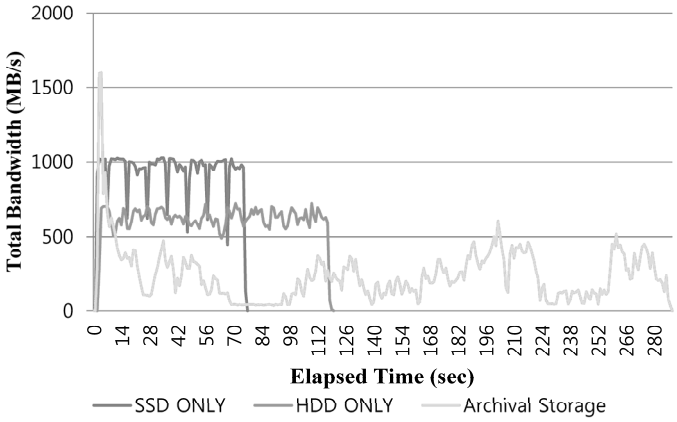

그림 2. 정책 별 질의 6번 수행 중 사용 대역폭 변화량

SSD ONLY의 경우에는 대역폭이 일정하게 1000MB/s로 유지되는 것을 볼 수 있으며, HDD ONLY의 경우 평균 650MB/s 정도로 유지되는 것을 확인할 수 있다. 그러나, Archival Storage의 경우 대역폭이 크게 요동치는 것(Bandwidth Fluctuation)을 확인할 수 있다. 이러한 급격한 대역폭 변동이 질의 성능을 저하시키는 주요한 원인이라고 볼 수 있다. 앞서 언급한 바와 같이 질의 6번의 3가지 모든 실험에서 CPU는 병목을 일으키지 않는다.

그렇다면 이러한 대역폭 변동은 어떤 이유에서 나타나는 것인가? 이는 각각의 디스크 간의 데이터 분배의 불균형으로부터 기인하며, 다음 섹션에서 자세하게 다룬다.

3.3. 디스크 간 블록 저장 비율

섹션 2에서 언급한 바와 같이, One_SSD 정책은 replication factor를 n이라고 했을 때, 하나의 블록은 SSD에 저장하고, 나머지 n-1개의 블록을 디스크에 저장하는 방식이다.

그림 3은 One_SSD 정책에서 각 노드의 SSD와 HDD 각각에 HDFS 블록이 저장된 양을 나타낸다. 본 실험의 Replication factor가 3이고, DataNode가 3대의 서버에

설정되었으므로 1:2의 비율로 저장되는 것을 확인할 수 있다. 그러나 각각의 디스크에 저장된 블록의 개수는 그림 4에서 보는 바와 같이 크게 다르다.

그림 3. HDD 및 SSD 간 데이터 저장 비율

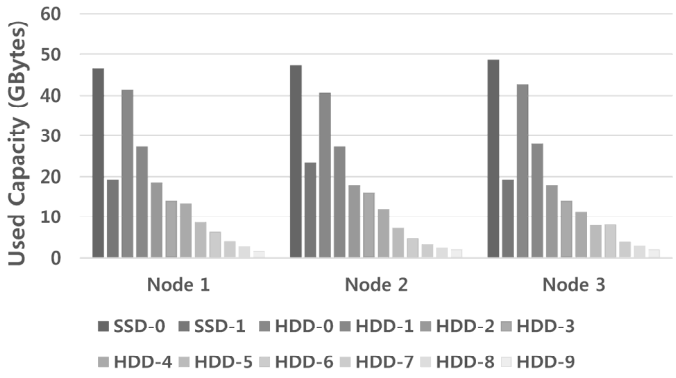

그림 4. 디스크 별 블록 저장 비율

SSD의 경우, 각 노드의 SSD-0에 평균적으로 약 70GB의 블록이 저장되고, SSD-1에 30GB 정도의 데이터가 저장되는 것을 확인할 수 있다. HDD의 경우는 그 차이가 더 큰데, 가장 많은 블록을 저장하는 경우가 약 60GB이며, 가장 적게는 3GB정도에 해당한다. 디스크 간 데이터 저장 불균형은 특정 디스크에 I/O 요청이 집중되는 결과를 초래하게 된다. 그러는 동안 다른 디스크의 대역폭을 활용할 수 있음에도 불구하고, 전체 대역폭을 활용하지 못하는 결과로 이어진다. 이는 다음 섹션의 실험 결과를 통해 확인할 수 있다.

3.4. 질의 수행 시 디스크 간 읽기 요청 불균형

그림 5는 질의 6번을 수행하는 동안 실제로 읽힌 블록의 총량을 디스크 별로 나타낸다. 그림 4에서 저장된 데이터의 불균형으로 인해 실제로 읽게 되는 데이터가 특정 디스크에 더 많이 집중되는 것을 확인할 수 있다. 이로 인해, 질의 수행 과정에서 특정 디스크에 집중된 데이터를 처리하는 경우 그림 2에서 보는 바와 같이 대역폭이 급격하게 변동하는 것을 확인할 수 있다. 결과적으로, 전체 대역폭을 활용하지 못하게 되면 대용량 데이터에 접근하는 질의의 경우 I/O 요청이 병목이 되어 질의 수행시간이 길어지는 문제가 발생하게 된다.

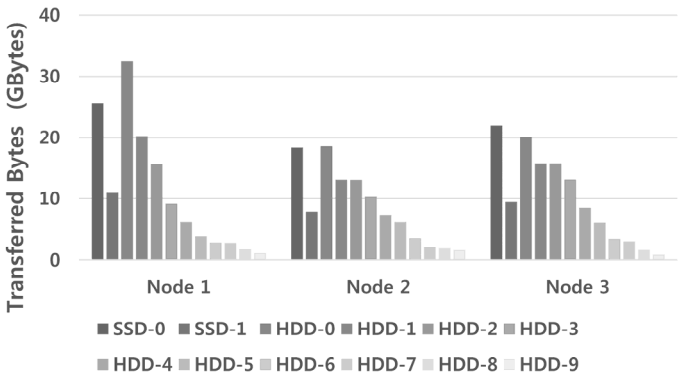

그림 5. 질의 수행 시 디스크 별 읽은 블록 총량

4. 결론 및 향후 과제

본 논문에서는 대표적인 빅데이터 플랫폼인 SparkSQL을 이용한 TPC-H 벤치마크 테스트를 통해 하둡의 분산 파일 시스템인 HDFS의 Hybrid array 기법인 Archival storage의 데이터 저장 정책이 갖는 문제점을 지적하고 있다. 결론적으로, 빅데이터 플랫폼에서 노드 내의 Task parallelism을 극대화시키기 위해서는 전체 저장장치의 대역폭을 고르게 사용하기 위한 디스크 간 데이터 저장량을 조절하는 정책이 필요하며, 뿐만 아니라 어떤 데이터를 SparkSQL 등의 Computation framework에 최적의 비율로 전달할 것인지에 대한 연구가 필수적이다.

향후에는 본 논문의 실험결과로부터 얻은 함의에 기반하여, Hybrid array를 빅데이터 플랫폼에서 효율적으로 활용하기 위한 데이터 분배 정책에 대한 연구를 수행할 예정이다.

참 고 문 헌

[1] SUBHLOK, J., et al., 1993. Exploiting task and data parallelism on a multicomputer. In ACM SIGPLAN Notices ACM, 13-22.

[2] MICHELONI, R., et al., 2013. Hybrid storage. In Inside Solid State Drives (SSDs) Springer, 61-77.

[3] Apache Hadoop. http://hadoop.apache.org

[4] ARMBRUST, M., et al., 2015. Spark sql: Relational data processing in spark. In Proceedings of the 2015 ACM SIGMOD International Conference on Management of Data ACM, Melbourne, VIC, Australia, 1383-1394.

[5] Archival Storage, SSD & Memory. https://hadoop.apache.org/docs/current/hadoop-project-dist/hadoop-hdfs/ArchivalStorage.html

[6] TPC-H. http://www.tpc.org/tpch/

[7] WANG, H., et al., 2014. Balancing fairness and efficiency in tiered storage systems with bottleneck-aware allocation. In Usenix Conference on File and Storage Technologies, Santa Clara, CA, 229-242.