

빅데이터 시스템 처리 엔진에 관한 성능 비교 연구

신민규*, 김태원*, 황선희*
 *kt NexR
 e-mail : jose.shin@nexr.com

Performance Comparison Study on Big Data System Processing Engine

Min-kyu Shin*, Tae-Won Kim*, Sun-Hee Hwang*
 *KT NexR

요약

최근 들어 빅데이터 처리 속도에 대한 관심이 높아지면서 처리 성능을 높이기 위한 다양한 기술들이 개발되고 있다. 기존의 빅데이터 전문 업체들도 전통적인 빅데이터 처리 방법이 아닌 새로운 기술에 대한 투자를 통해 경쟁을 가속화 하고 있다. 이러한 이유로 최적의 처리 엔진을 선택하는 것은 점차 중요한 부분이 되고 있다. 하지만 실제로 환경을 구성하여 운영해보기 전까지는 시행착오를 겪을 수 밖에 없는 상황이다. 그래서 본 논문에서는 최근 발표된 빅데이터 처리 엔진 기술들이 어느 정도 성능을 보이는지에 관한 연구를 수행하여 빅데이터 처리 엔진 선택 및 판단에 도움이 될 근거를 제공하고자 한다.

1. 서론

빅데이터에서 원하는 정보 및 의미 있는 데이터를 찾으려는 데이터 가공 및 분석 과정을 지원하는 것이 빅데이터 처리 기술이다. 빅데이터 처리 환경에서는 확장성, 데이터 생성 및 처리 속도를 해결하기 위한 처리시간 단축 및 실시간 처리 지원, 비정형 데이터 처리 지원 기술 등이 필요하다. 대표적으로 Hadoop[1]은 분산 파일 시스템인 HDFS 와 맵리듀스로 구현된 빅데이터 처리 기술이다. 그리고 최근 들어 빅데이터 처리 속도에 대한 관심이 높아지면서 처리 성능을 높이기 위한 다양한 기술들이 개발되고 있다.

이에 따라 DW 및 RDBMS 시장의 중심에 있던 많은 기업들의 변화가 포착되고 있다. 또한 기존의 빅데이터 전문 업체들도 전통적인 빅데이터 처리 방법이 아닌 새로운 기술에 대한 투자를 통해 경쟁을 가속화 하고 있다. 이에 따라 최적의 처리 엔진을 선택하는 것은 점차 중요한 부분이 되고 있다. 많은 업체들이 자신의 솔루션 성능에 대한 테스트 결과에 대해 발표하고 있지만 해당 솔루션에 최적화된 소프트웨어, 하드웨어 환경에서 측정된 일방적인 결과들이 많다.

이러한 이유로 필요한 처리 엔진 선택함에 있어 실제로 환경을 구성하여 운영해보기 전까지는 시행착오를 겪을 수 밖에 없는 상황이다.

그래서 본 논문에서는 최근 발표된 빅데이터 처리 엔진 기술들이 어느 정도 성능을 보이는지에 관한 연구를 수행하여 빅데이터 처리 엔진 선택 및 판단에 도움이 될 근거를 제공하고자 한다. 본 논문에서는 빅데이터 처리 엔진인 Apache Hive[2], Facebook

Presto[3], Apache Phoenix[4] 의 성능을 동일한 하드웨어 환경에서 비교하고, 추가적으로 기존에 널리 사용되고 있는 RDMS 인 MariaDB[5] 성능과 비교하여 어느 정도 차이가 나는지에 관한 연구를 수행하고자 한다.

2. 관련 연구

2.1 Apache Hive

아파치 하이브(Apache Hive)는 하둡에서 동작하는 데이터웨어하우스(Data Warehouse) 인프라 구조로서 데이터 요약, 질의 및 분석 기능을 제공한다. 초기에는 페이스북에서 개발되었지만 넷플릭스 등과 같은 회사에서 사용되고 있으며 개발되고 있다.

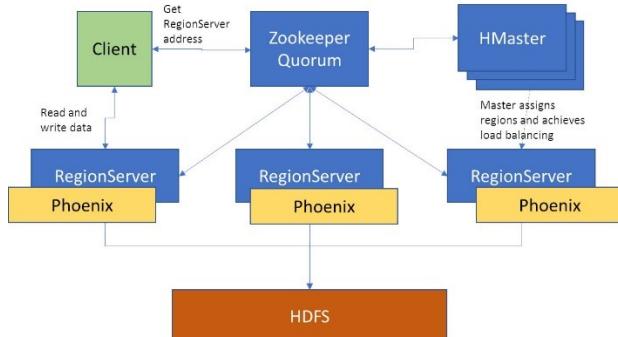
아파치 하이브는 HDFS 나 아파치 HBase[6]와 같은 데이터 저장 시스템에 저장되어 있는 대용량 데이터 집합들을 분석한다. HiveQL[7] 이라고 불리는 SQL 같은 언어를 제공하며 맵리듀스의 모든 기능을 지원한다. 쿼리를 빠르게 하기 위해 비트맵 인덱스를 포함하여 인덱스 기능을 제공한다.

기본적으로 하이브는 메타데이터를 내장된 아파치 더비(Derby) 데이터베이스 안에 저장한다. 그렇지만 MySQL 과 같은 다른 서버/클라이언트 데이터 베이스를 사용할 수 있는 선택권을 제공한다. 현재 TEXTFILE, SEQUENCEFILE, ORC 그리고 RCFILE 등 4 개의 파일 포맷을 지원한다.

2.2 Apache Phoenix

RDBMS 가 수십년 동안 인기가 있었고, 최근에는 전

통적인 관계형 데이터베이스 보다 덜 제한적인 일관성 모델을 이용하여 데이터의 저장 및 검색을 위한 매커니즘을 제공하는 NoSQL 데이터베이스라고 불리는 데이터베이스가 나타나기 시작했다. 초기에 NoSQL 데이터베이스가 출시되었을 때는 SQL 인터페이스는 지원하지 않았다. 하지만 시간이 지나면서 기존 NoSQL 데이터베이스에 SQL 인터페이스를 지원하는 솔루션들이 개발되었다.



예를 들어 HiveQL은 SQL 문과 유사한 언어를 제공하여 HDFS에 저장된 데이터를 조회할 수 있는 인터페이스를 지원하기 위해 만들어졌다. 이와 비슷하게 (그림 1)과 같은 구조로 HBase에 SQL 인터페이스를 지원하기 위해 나온 것이 Apache Phoenix다.

Phoenix는 SQL로 HBase 데이터에 접근할 수 있게 구현된 오픈 소스이다. HBase 클라이언트 API를 사용하는 대신 표준 JDBC API를 사용하고, Spark, Hive, Pig, Flume, MapReduce 같은 다른 하둡 제품들과 잘 연동된다.

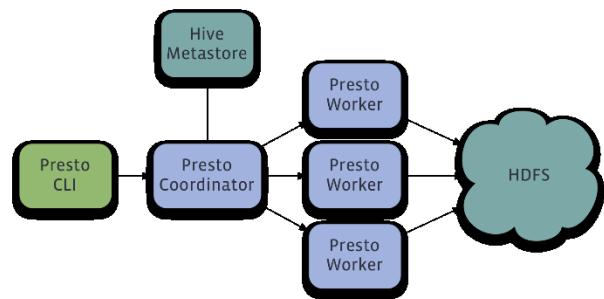
그리고 아파치 HBase는 Hadoop 플랫폼을 위한 비관계형 분산 데이터 베이스이다. 구글의 빅테이블(BigTable)[8]을 본보기로 삼았으며 자바로 쓰여졌다. 아파치 소프트웨어 재단의 아파치 하둡 프로젝트 일부로서 개발되었으며 hadoop의 분산 파일 시스템인 HDFS 위에서 동작 한다.

2.3 Facebook Presto

프레스토(Presto)는 페이스북이 개발한 빅데이터 분석 도구로, 상호작용 속도에 맞춰진 애드-혹 분석에 최적화된 분산된 SQL 쿼리 엔진이다. 복잡한 쿼리들, 집계, 조인들과 윈도우 함수들이 포함된 표준 ANSI SQL을 지원한다. 기존 분석 도구인 하이브/맵리듀스에 비해 CPU 효율성과 대기 시간이 10배 가량 빠르며, '최소 비용으로 효율적인 컴퓨팅 인프라를 구축'하자는 오픈 컴퓨터 프로젝트의 일부로, 2013년 11월 7일 아파치 라이선스로 공개되었다.

프레스토는 자바로 만들어졌으며, (그림 2)와 같이 Coordinator와 Worker 노드를 통해 메모리 처리와 데이터 구조 기술을 적절히 혼합하여 자바 코드의 메모리 할당과 쓰레기 정보 수집 문제를 피할 수 있도록 설계 되었다. 또한 HDFS와 HBase와 같은 데이터 저장소 와도 쉽게 연결 할 수 있도록 다양한

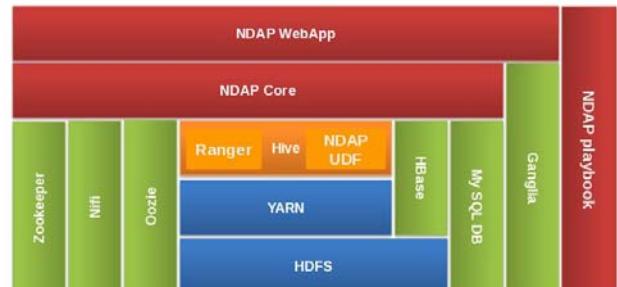
Connector들을 제공한다.



(그림 2) Facebook Presto Overview

2.4 NDAP과 Presto/Phoenix Connector

NDAP[9]은 빅데이터 전문기업인 kt NexR에서 개발한 빅데이터 소프트웨어 플랫폼이다. NDAP은 빅데이터 활용에 필요한 데이터 수집, 저장, 분석, 검색, 변환 등을 모두 제공하는 소프트웨어 플랫폼으로서 글로벌 빅데이터 표준기술로 부상한 Hadoop의 주요 핵심 기능들을 표준화하여 (그림 3)과 같은 아키텍처로 엔터프라이즈 환경에서 최적화된 성능, 신뢰성, 사용편의성을 제공한다.



(그림 3) kt NexR NDAP Architecture

그리고 NexR Presto/Phoenix Connector는 Facebook Presto가 Phoenix를 통해 HBase를 저장소로 사용할 수 있게 연결하고, Hive Connector를 사용할 때, Update를 지원하지 않는 한계점을 해결하기 위해 UPSERT를 할 수 있도록 NexR에서 개발한 Plugin이다.

3. 성능 비교

3.1 성능 비교 환경

빅데이터 처리 엔진들의 성능을 측정하기 위한 하드웨어 환경은 네임 노드 및 NDAP 그리고 기타 Hadoop Ecosystem들을 설치하기 위한 관리 노드 4대와 데이터 노드 8대로 구성하였으며, 자세한 내용은 아래 <표 1>과 같다.

동일한 환경에서의 성능 확인을 위해 각 엔진들의 기본 설정 값을 유지한 상태에서 테스트를 진행하고자 했으며, 각 엔진들의 최적의 메모리 활용을 위해 Hadoop 맵리듀스, HBase Region-Sever, Facebook

Presto 의 Heap 메모리 크기는 하드웨어 환경에 맞게 아래와 같이 수정하여 구성하였다.

- Hadoop MR Heap Size : 32 GB
- HBase Region-server Heap Size : 32 GB
- Facebook Presto Worker Heap Size : 32 GB

<표 1> 하드웨어/소프트웨어 환경

# of Mgmt Node	4 대
# of Data Node	8 대
CPU	16 core (Intel Xeon Silver 4110, 2.10 GHz)
RAM	128G
DISK	SAS 4TB * 4
OS	CentOS release 6.9 (Final)
NexR NDAP	4.10
HDFS	2.7.3
Hive	2.3.2
HBase	1.2.6
Presto	0.218
MariaDB	10.1

3.2 성능 비교 방법

효과적인 처리 엔진 성능 비교를 위해서 대량의 데이터 시험 및 성능 시험에 사용되는 벤치마크는 TPC-H 를 사용하였다. TPC-H 벤치마크 테스트[10]는 의사 결정 지원 시스템의 성능 테스트를 위해 개발된 것으로, 주로 의사 결정 용도의 시스템에 대한 성능 측정을 위해 참조되며, Business 를 위한 Ad Hoc Query 위주로 구성된다. TPC-H 테스트 단위는 QphH@Size(TPC-H Composite Query per Hour)이며, TPC-H 측정방법은 8 개의 Table 로 이루어진 데이터 베이스에 대해 22 개의 사전 정의된 Query 를 동시에 수행하여 소요 시간을 측정한다. 측정된 소요시간은 정의된 공식에 따라 Raw data 의 크기, Stream 수 등과 함께 <표 2> 와 같이 계산되어 표현한다. 본 논문에서는 TPC-H 2.18.0 버전을 사용하였다.

<표 2> TPC-H 테스트 단위[10]

$$\begin{aligned} \text{QphH@Size} &= \sqrt{\text{Power@Size} * \text{Throughput@Size}} \\ * \text{Power@Size} &= 3600 * \exp \left\{ -\frac{1}{24} \left[\sum_{i=1}^{i=22} \ln(QI(i, 0)) + \sum_{j=1}^{j=2} \ln(RI(j, 0)) \right] \right\} * SF \\ * \text{Throughput@Size} &= (S * 22 * 3600) / T_s * SF \end{aligned}$$

그리고 Hive 의 Transaction 테이블 기능과 Phoenix 의 UPSERT 기능을 활용하여 빅데이터 처리 엔진들이 전통적인 RDBMS 엔진 과도 동일 조건에서 성능을 비교할 수 있도록 하였다.

3.3 성능 비교 결과

각 엔진들의 성능 비교 결과는 <표 3> 과 같다. 가

장 좋은 성능을 나타낸 엔진은 Facebook Presto 이며, Hive 엔진이 가장 나쁜 성능을 보였다. 특히 Hive 의 경우, Transaction 기능을 사용하기 위해서 생성할 테이블에 컬럼 해시를 기준으로 데이터를 지정된 개수의 파일로 분리해서 저장하는 Bucket 설정이 필수인데, Bucket 을 16 으로 설정 시(Hive-tr16), 과도한 파일 생성 및 맵태스크가 생성되어 매우 큰 성능 저하로 연결되는 것을 확인하였다. Bucket 을 1 로 (Hive-tr1) 설정하여도 non-Transaction 테이블보다 절반 이하의 성능저하가 있는 것으로 볼 때, Hive Transaction 테이블은 non-Transaction 테이블보다 성능이 불리하고, Hive 트랜잭션 테이블 생성 시 테이블 사이즈에 맞게 Bucket 사이즈를 고려해야 함을 확인하였다.

TPC-H 는 데이터 모델이 여러 index 를 기반으로 설계되어 있어, Phoenix 테스트 환경에서는 Primary Index 만 생성 후 테스트를 진행하였다. 이로 인해 인덱싱이 적용되지 않은 컬럼이 조건절에 있을 때는 Full-Scan 이 되어 <표 4> 에서와 같이 성능이 저하되는 것을 확인하였다.

Presto 는 가장 좋은 성능을 보였고, RDBMS 인 MariaDB 보다도 더 좋은 결과를 나타냈다. 비록 Transaction 처리를 위해서는 Presto/Phoenix Connector 가 필수이지만 추후 RDBMS 를 대체할 수 있는 성능을 가지고 있음을 확인하였다.

4. 결론

빅데이터 처리 엔진들이 다양한 형태의 대용량 데이터를 효율적으로 처리하고 분석하는 기술로 채택되고 있지만 아직도 많은 분야에서 관계형 데이터베이스 시스템을 이용하여 데이터 분석을 수행하고 있다.

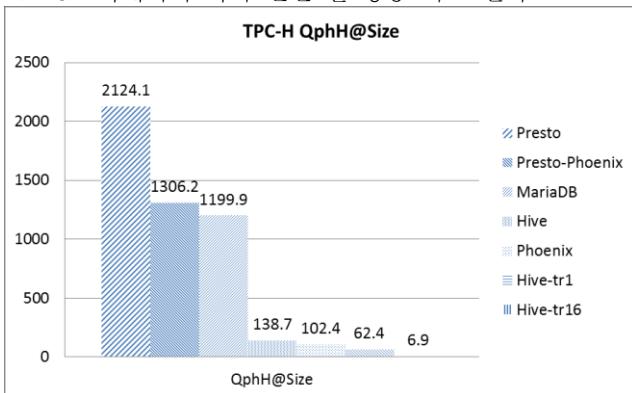
이는 빅데이터 처리 엔진들의 성능이 관계형 데이터베이스에 비해 얼마나 나은 결과를 보이는지에 관한 연구가 많지 않으며 운영 시스템에 적용하기 위해서는 많은 비용과 시행착오가 필요하기 때문이다.

본 연구에서는 가장 많이 사용되고 있는 빅데이터 처리 엔진인 Hive, Phoenix/HBase, Presto 에 대해 성능 비교를 수행하여 빅데이터 처리 엔진 선택 및 판단에 도움이 될 근거를 제공하고자 하였다.

성능을 비교한 결과에서는 메모리 처리와 데이터 구조 기술을 적절히 혼합하여 사용하는 아키텍처를 가지고 있는 Facebook Presto 가 RDBMS 인 MariaDB 를 포함하여 다른 처리 엔진보다 좋은 데이터 처리 성능을 보였다. 이는 빅데이터 처리 엔진을 환경과 상황에 맞게 알맞은 전략을 세워 빅데이터 처리를 수행한다면 관계형 데이터베이스 보다 나은 성능을 얻을 수 있음을 본 연구에서 실험적으로 보인 것이다. 기존의 일부 RDBMS 도 빅데이터 처리가 가능해 졌지만 오픈소스 기반의 아키텍처 도입으로 적은비용으로 투자대비 큰 효과를 낼 수 있는 장점도 존재한다. 빅데이터 처리 엔진은 다양한 업체와 더 많은 개발자들이 개발에 참여하여 다양한 기능이 추가되고 안정성이 개선될 것으로 보인다.

이처럼 점차 기술의 완성도가 높아져 빅데이터 분석에서 처리 엔진의 선택은 중요한 축을 담당할 것으로 예상된다. 그러므로 향후에도 빅데이터 처리 엔진들에 대한 지속적인 연구가 진행된다면 기존의 DW 와 RDBS 시장에서도 활용도가 더욱 높아질 수 있을 것으로 예상된다.

<표 3> 빅데이터 처리 엔진 별 성능 비교 결과



<표 4> 빅데이터 처리 엔진 별 TPC-H 수행 결과

	Presto	MariaDB	Phoenix	Hive-tr1
q1	92	65.5	27.6	361.9
q2	5.7	2.3	1,006.50	149.1
q3	9.8	5.7	125.3	227.4
q4	9.9	3.6	282.2	440.1
q5	10.3	4.8	187.9	455.6
q6	2.6	11.9	20.2	57.2
q7	12.2	3.8	27.3	222.3
q8	11.3	6.8	1,735.30	517.5
q9	15.8	102.4	51.3	629.4
q10	9.5	23.4	69.7	419.1
q11	4.7	0.7	5.2	128.7
q12	14.5	30.3	20.9	372.2
q13	8.2	17.1	107.1	86.2
q14	1.9	201.8	29.1	44.6
q15	3.1	26	31.9	100.6
q16	2.1	6.6	268.6	106.6
q17	9.2	0.4	90.9	786.5
q18	16.7	23.5	544.6	682.2
q19	8.7	0.7	1,810.00	372.1
q20	3.5	3.8	65.7	157
q21	25.7	58	27.3	1087
q22	11.8	0.8	31.7	106.4
QphH @Size	2124.1	1199.9	102.4	62.4

참고문헌

- [1] Apache Hadoop [Internet], <https://hadoop.apache.org/>
- [2] Apache Hive [Internet], <https://hive.apache.org/>
- [3] Facebook Presto [Internet], <http://prestodb.github.io/>
- [4] Apache Phoenix [Internet], <https://phoenix.apache.org/>
- [5] MariaDB [Internet], <https://mariadb.org>
- [6] Apache HBase [Internet], <https://hbase.apache.org/>
- [7] Apache HiveQL [Internet],
https://docs.hortonworks.com/HDPDocuments/HDP3/HDP-3.1.0/using-hiveql/content/hive_hive_query_language_basics.html
- [8] Google Bigtable [Internet],

<https://cloudplatform.googleblog.com/2015/05/introducing-Google-Cloud-Bigtable.html>

- [9] kt NexR NDAP [Internet],
<https://www.ktnexr.com/product/ndap.html>
- [10] TPC Benchmark™H (TPC-H) [internet],
<http://www.tpc.org/tpch/>
- [11] Choong-Hyun Yoo, "Technology Trends in Big Data Analytics and Introduction to R," NexR, 2012
- [12] Kwang-Man KO, Beom-Chul Kwon, Sung-Chul Kim, Sang-Jun Lee, "Development of Statistical Prediction Engine for Integrated Log Analysis Systems," Journal of The 2013 Fall Conference of the KIPS, Vol. 20, No. 2, 2013.
- [13] An Oracle White Paper, "Hadoop and NoSQL Technologies and the Oracle Database", February 2011
- [14] T. G. Armstrong, V. Ponnekanti, D. Borthakur, M. Callaghan, K. A. Ross, D. Srivastava, D. Papadias, "Linkbench: A database benchmark based on the facebook social graph", Proc. ACM SIGMOD Int. Conf. Manage. Data, pp. 1185-1196, Jun. 2013
- [15] J. Dean, S. Ghemawat, "MapReduce: Simplified data processing on large clusters", Commun. ACM, vol. 51, no. 1, pp. 107-113, 2008