

# 빅데이터 하둡 플랫폼의 활용

이현종

(주)빅스터

## 요약

인터넷의 활성화 및 모바일 서비스의 등장으로 빅데이터 시대를 맞이하게 되었다. 이전에는 저장 및 처리할 수 없었던 영역, 이제는 새로운 기술의 등장과 분석을 통한 가치 창출의 가능성으로 빅데이터는 IT 업계의 최대 화두가 되어 가고 있다. 이러한 빅데이터를 바라보는 시각은 크게 기술적 관점과 분석적 관점으로 나뉘고 있다. 특히 기술적 관점에서 바라보는 빅데이터는 하둡을 표준으로 하는 오픈소스 분석 플랫폼의 대두가 고무적이다. 누구나가 대용량의 확장 가능한 시스템을 운영할 수 있는 기회가 온 것이다. 본 고에서는 빅데이터의 그 태생적 특징을 살펴보고, 비교적 저렴한 비용의 플랫폼 환경 구축을 위해 오픈소스 하둡이 널리 활용되고 있는 이유에 대해 알아본다. 또한 하둡의 용도와 어떠한 종류의 데이터 분석을 위해 사용되어지고 있는지, 그리고 하둡의 구성 및 하둡 생태계를 이루고 있는 요소들이 무엇인지 살펴본다. 끝으로 빅데이터를 활용하기 위한 6단계 절차와 이에 발맞춰 하둡 플랫폼을 어떻게 효율적으로 활용할 지에 대해 그 방법을 모색해 보고자 한다.

## I. 서론

1990년대 말 인터넷의 활성화로부터 촉발된 디지털 데이터의 생산은 2000년대 중반을 거쳐 그 양을 가늠할 수 없게 되었다. 개인 블로그 미디어의 인기 이후 등장한 페이스북, 트위터 등 SNS, 언제 어디서든 접속 가능한 모바일 서비스 등은 빅데이터 시대가 도래하는 원동력이 되었다 [1].

빅데이터를 바라보는 시각은 크게 두 부류로 나뉜다. 하나는 데이터 처리의 방법을 다루는 '기술', 그리고 다른 하나는 가치를 창출하기 위한 '분석'이 그것이다. 빅데이터와 관련된 현업에서 종사하는 사람들은 기술을 우선으로 할 것인지 아니면 분석을 우선으로 할 것인지에 대해 첨예한 대답을 보이고 있다. 그 배경에는 빅데이터가 기술로부터 잉태되었다는 숨은 사실이 있다.

인터넷의 대중화는 수많은 웹문서의 생성을 야기했고, 정보의 홍수속에서 가장 빠르고 정확한 결과를 얻기 위한 수단으로 검색엔진이 등장하였다. 야후의 1세대 검색을 거쳐 구글이라고 하는 초유의 웹검색엔진이 탄생하였다. 전세계의 모든 웹문서를 수집하고, 페이지랭크(Page Rank)라고 하는 특출난 가치치 부여 알고리즘을 통해 구글은 최고의 검색엔진으로 자리잡았다.

2000년대 초반 구글의 고민은 무한히 생성되는 웹문서를 기존의 시스템과 달리 비교적 저렴하게 수집 및 저장하고, 2개월 이상 걸리던 페이지랭크 계산 시간을 최단 기간으로 단축하고자 하는 것이었다. 또한 전세계 네트워크에 검색서비스를 공급하기 위해서는 시스템의 안정성 역시 보장되어야 했다. 이를 해결하기 위해 나온 것이 바로 분산 저장 방식을 채택한 '구글 파일시스템(GFS)'과 대규모 병렬 처리 프레임워크인 '맵리듀스(MapReduce)'였다.

기존의 시스템과 아키텍처로는 처리하지 못했던 빅데이터. 속명적인 필요의 노력 끝에 저장하고 처리할 수 있는 능력이 부여된 것이다. 2003년에 발표된 구글의 논문[2]에 기인하여 '하둡(Hadoop)'이라고 하는 신출귀몰한 오픈소스 프로그램이 나오게 되었고, 이와 병행하여 야후와 아마존, 페이스북 등 웹서비스 절대 강자들로부터 다양한 데이터 처리 기술들이 쏟아져 나왔다.

본 고에서는 이러한 빅데이터의 활용 가치와 이를 효율적으로 분석하기 위한 오픈소스 하둡 플랫폼의 전반에 대해 살펴보고, 빅데이터 분석 플랫폼으로서의 하둡이 가지는 장점을 살펴 안정적이고 원활한 운영 방안에 대해 모색해 본다.

## II. 빅데이터 활용을 통한 가치 창출

우리가 접하고 있는 빅데이터의 활용 가치는 크게 새로운 비즈니스의 기회와 공익적 목표 달성에 있다. 빅데이터 분석을 통해 자사의 수익을 극대화 하거나, 제조비용 또는 생산원가를 절감할 수 있는 시도가 진행되고 있다. 분석된 데이터를 통해 마

케팅에 활용하거나 과학자들의 연구 개발 활성화에도 큰 도움이 되는 추세이다. 공공분야에서의 빅데이터는 국민들의 편의 및 안전 보장, 건강 증진 등을 위한 좋은 재료가 되고 있다.

이러한 빅데이터의 속성으로 크게 3V를 언급하곤 한다. 3V란 크기(Volume), 속도(Velocity), 다양성(Variety)을 이른다. 그러나 빅데이터의 활용적 측면에서 바라보았을 때 중요한 속성 중 하나가 바로 '저비용(Low Cost)'이다. 빅데이터의 태생적 근원이 기존의 그것과는 다른 체계에서 비교적 저렴한 비용으로 저장되고 처리되길 바라는 것이었기 때문이다[3].

실제로 국내에서는 다양한 업체들의 빅데이터 플랫폼 전쟁이 벌어지고 있는 형국이며, 누가 주도권을 잡을지는 아무도 모른다. 다만 확실한 것은 빅데이터의 태생상 저비용을 만족시켜야 하기 때문에 이를 활용하여 얻을 수 있는 이익보다 비용이 많이 들어서는 안 된다는 사실이다. 사람들은 이러한 관점으로 인해 오픈소스에 눈길을 주고 있으며 그 중심에는 하둡이 서 있다.

### Ⅲ. 오픈소스 하둡의 역사



그림 1. 오픈소스 하둡의 공식 로고[4]

하둡(Hadoop)은 대용량 데이터 처리를 위해 거대한 컴퓨터 클러스터에서 동작하는 분산 응용 프로그램을 지원하는 오픈소스 프레임워크이다. 원래 오픈소스 웹검색엔진 너치(Nutch)의 분산처리를 지원하기 위해 개발된 것으로, 아파치 루씬(Lucene)의 하부 프로젝트로 시작되었다. 구글파일시스템(GFS)을 벤치마킹하여 하둡분산파일시스템(HDFS:Hadoop Distributed File System)과 맵리듀스(MapReduce)를 구현한 것이다.

그 역사[5]를 보면 하둡은 아파치 루씬의 창시자인 더그 커팅(Douglas Cutting)에 의해 시작되었다. 더그 커팅은 2003년에 있는 구글의 분산 파일 시스템 아키텍처 논문발표를 보고 2004

년에 NDFS를 개발하였다. 2004년에 발표된 역시 구글의 맵리듀스 관련 논문에 기인하여 2005년 중반까지 NDFS에 맵리듀스를 반영하게 되었다. 2006년 2월에 NDFS와 맵리듀스는 하둡이라는 이름으로 너치에서 별도의 서브프로젝트로 빠지게 된다. 이 즈음 더그커팅은 야후에 합류하게 되고 2008년 2월, 결국 하둡은 아파치 최고 수준의 프로젝트로 등극하게 되었다.

그 성능을 보자면 2008년 4월에 있었던 테스트에서는 테라바이트 이상의 데이터 정렬을 위해 가장 빠른 시스템으로서 세계 기록을 갱신하였고, 2009년 5월에 야후는 하둡을 이용하여 62초만에 1테라바이트 데이터를 정렬하는 쾌거를 올렸다.

그렇다면 하둡은 어디에 사용되어질까. 하둡은 크게 4가지 용도를 가지고 있다. 첫째는 검색엔진 색인저장소(Indexing), 둘째는 데이터 분석 또는 통계분석(Analytics/Statistical Analytics)이다. 셋째는 데이터의 전처리(Table Precomputation and Rollup)이고 넷째는 정형 데이터의 저장소(Structured Data Storage)이다.

하둡이 구글의 검색엔진 시스템을 본딴 것이기 때문에 색인저장소로 쓰이는 데에는 이견이 없다. 그러나 사람들은 분산 저장의 특징 및 병렬 처리의 성능을 확인하고 하둡을 빅데이터 분석 플랫폼으로 인식하게 되었다. 물론 기존의 고가의 장비를 앞세운 스트리저보다 더욱 안정적이고 월등한 저장소의 역할도 하고 있다.

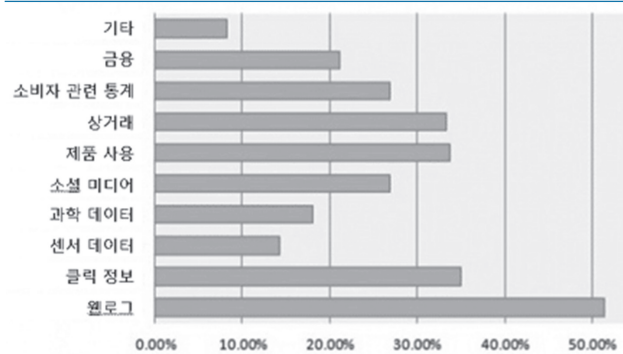


그림 2. 하둡 활용 데이터 종류

사람들은 어떤 데이터의 분석이나 저장을 위해 하둡을 사용하고 있는지 지난 6월 하둡 전문 기업 Karmasphere에서 376명의 데이터 전문가와 인터뷰한 결과[6]가 있다. <그림 2>를 보면 웹로그 분석이 단연 으뜸이고 그 다음이 클릭 정보와 상거래 데이터였다. 역시 인터넷 서비스 분야의 데이터가 많았다. 소셜 미디어 또는 금융 및 CRM 관련 고객 데이터 분석에도 많이 쓰이고 센서 데이터를 포함한 과학 데이터를 위해서도 하둡은 사용되고 있었다.

그렇다면 이런 빅데이터 시대에 왜 하둡이 각광을 받는 것인가. 모든 글로벌 벤더들이 자사의 IT 제품에 하둡과의 연결고리

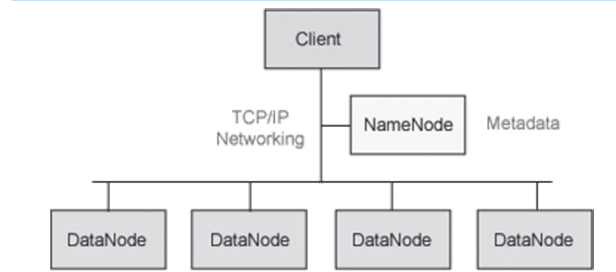
를 만들 정도이니 그 인기는 하늘을 찌르고 있는 상태이다. 하둡이 빅데이터 분석 플랫폼의 표준이 되어 가고 있는 이유는 무엇일까. 빅데이터에 있어 하둡이 적합한 이유는 크게 아래의 네 가지로 요약된다.

첫째, 애플리케이션 및 트랜잭션 로그 정보는 매우 크다. 이를 해결하기 위해 하둡은 대용량 파일을 저장할 수 있는 분산 파일 시스템을 제공한다. 하둡의 DFS는 대용량의 비정형 데이터 저장 및 분석에도 효율적이다. 둘째, 기존 시스템은 I/O 집중적이면서 CPU도 많이 사용한다. 단일 서버를 단순 연결할 경우 최대의 성능을 기대하기는 어렵다. 이에 반해 하둡은 클러스터 구성을 통해 멀티 노드로 부하를 분산시켜 처리한다. 개별적인 서버에서 진행되는 병렬처리 결과를 하나로 묶어 시스템의 과부하나 병목 현상을 줄였다. 셋째, 데이터베이스는 하드웨어를 추가한다고 하더라도 성능 향상이 선형적이지 않다. 하둡은 장비를 증가시킬 수록 성능이 linear에 가깝게 향상된다. 다수의 서버를 단일화된 클러스터로 구성하는 가장 큰 장점이다. 넷째, 데이터베이스는 소프트웨어와 하드웨어가 비싸다. 오픈소스 하둡은 무료이며, 이를 탑재하게 되는 Intel Core 머신과 리눅스 OS는 상대적으로 저렴하다. 어찌 보면 네번 째 이유가 가장 두드러지는 것일지도 모르겠다.

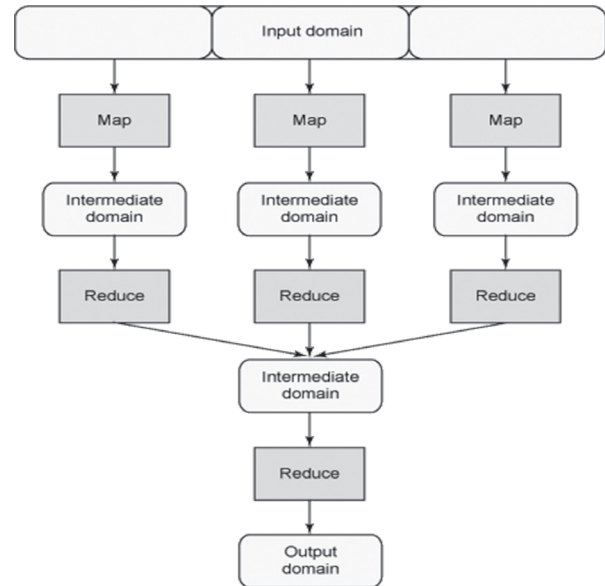
#### IV. 하둡 플랫폼의 구성

빅데이터를 위한 플랫폼의 정점은 대용량 데이터의 분산저장과 다수의 서버 클러스터에서 일어나는 병렬처리를 꼽을 수 있다. 하둡은 이에 최적화되어 있다. <그림 3(a)>를 보면 파일을 기본 64Mbyte 단위로 나누어 분산 저장하는 하둡분산파일시스템(HDFS, Hadoop Distributed File System)은 안정적이고 빠른 저장소의 역할을 한다. 이는 네임노드와 데이터노드로 분리되어 동작한다. 여러 요소로 인해 데이터노드에 장애가 발생하였을 경우 새로운 데이터노드를 추가하기만 하면 기존의 시스템을 유지시켜 주는 안정적 설계로 되어 있다.

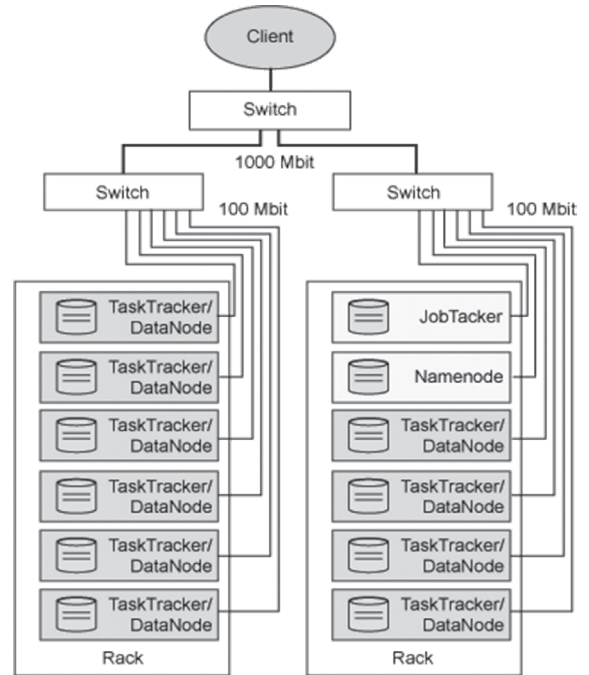
<그림 3(b)>를 보면 맵리듀스(MapReduce)는 대용량의 데이터를 빠르고 안전하게 병렬처리하기 위해서 보통의 상용 하드웨어(Commodity Hardware)를 이용한 분산 프로그래밍 모델이라고 할 수 있다. 기존의 IT 아키텍처가 애플리케이션이 있는 서버로 데이터를 불러와 처리를 하는 방식이었다면 하둡의 맵리듀스는 클러스터로 묶인 개별 데이터 노드로 프로그램을 보내 데이터가 존재하는 서버에서 직접 처리하는 사상으로 설계되어 있다. 이로 인해 맵리듀스를 통한 병렬처리는 실시간에는 어울리지 않고 일괄 배치(Batch) 작업에 최적화되어 있다.



(a) HDFS의 단순 개념도



(b) Mapreduce의 단순 개념도



(c) 하둡 클러스터의 물리적 분산 구조도

그림 3. 하둡 플랫폼의 개념 및 구조[7]



〈그림 3(c)〉에는 데이터 분석 처리 절차와 저장소로서의 하둡 클러스터를 물리적으로 분산한 예시가 나와 있다. 서버가 위치하는 랙(Rack)을 여러 개로 두어 전원장치 등의 문제로 인한 장애를 방지하고, 실제 데이터가 쌓이는 데이터노드와 병렬처리를 가능하게 하는 태스크트래커를 물리적으로 단일한 서버에 두어 효율을 극대화하는 모습을 볼 수 있다.

하둡을 구성하는 핵심은 하둡분산파일시스템과 맵리듀스이다. 그러나 오픈소스의 특성상 다양한 연관 솔루션과 도구들이 등장하는데 이러한 연관 관계를 하둡 생태계(Hadoop Ecosystem)라고 부른다. 즉, 사용자들은 하둡을 위해 별도의 솔루션 및 도구를 만들 필요 없이 하둡 생태계를 잘 활용하기만 한다면 어렵지 않게 손쉬운 빅데이터 분석 플랫폼을 구축할 수 있는 것이다.

사용자 입장에서는 하둡 생태계에 대한 올바른 이해와 그 연관성을 숙지하고 개별적인 솔루션을 이용할 필요가 있다. 그렇다면 하둡 생태계에 대해 자세히 알아보자.

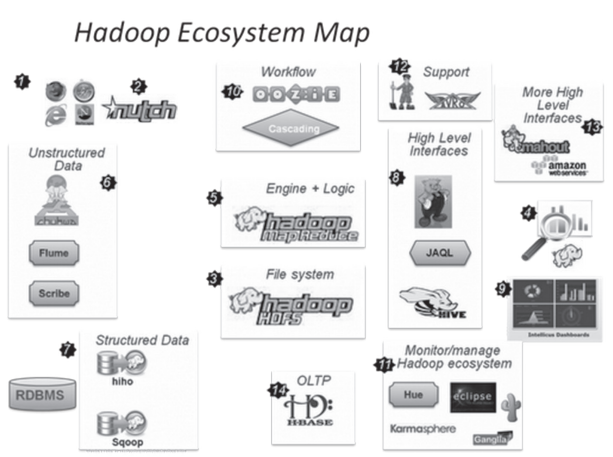


그림 4. 하둡 생태계 지도

〈그림 4〉의 하둡 생태계 지도를 번호 순서대로 설명하면 다음과 같다[8].

①번은 하둡의 시작을 알린다. 웹상에 존재하는 엄청난 양의 데이터는 3V의 Volume을 가능케 한다. ②번은 데이터를 수집하기 위한 솔루션이다. 초기 하둡의 상위 프로젝트로서 너치(Nutch)는 이런 웹상의 모든 데이터를 수집하기 위한 웹로봇으로 설계되어 있다.

③번에는 빅데이터의 저장을 위해 하둡분산파일시스템(HDFS)이 존재하게 되고, ④번에서는 이런 데이터를 활용할 수 있는 방법을 강구하게 된다. 맵리듀스 프레임워크(Map Reduce Framework)는 ⑤번에서 병렬 처리와 분석 실행을 위해 존재하며, 스트리밍 또는 파이프 방식으로 자바 및 다양한 언어들을 지원하고 있다.

웹로그와 클릭정보 등 일반적인 파일 형태의 데이터를 수집하기

위해서는 ⑥번의 퓨즈(Fuse), 웹다브(Webdav), 척와(Chukwa), 플룸(Flume), 스크라이브(Scribe) 등의 도구를 활용한다. 데이터베이스 형태로 존재하는 것도 있을 수 있는데 이는 ⑦번에서 하둡 분산파일시스템(HDFS)에 RDBMS의 정형 데이터를 넣기 위한 하이호(Hiho), 스콥(Sqoop) 등의 도구를 활용하면 된다.

수집된 데이터를 처리하기 위해서는 맵리듀스를 사용하게 되는데 ⑧번에는 고난도의 맵리듀스 프로그래밍을 간편하게 사용하기 위해 피그(Pig), 하이브(Hive), 자클(Jaql) 등의 도구가 있다. ⑨번의 인텔리커스(Intellicus) 드릴다운 등 향상된 UI 리포팅과 함께 하는 BI 툴이다. 맵리듀스 프로세스와 고수준 언어들을 위해 필요한 워크플로우 스케줄러인 우지(Oozie)와 캐스캐이딩(Cascading)은 ⑩번에 소개되어 있다.

⑪번에는 하둡의 모니터링 및 관리, Hive 등 여러 툴 운영, HDFS의 모니터링을 위한 휴(Hue), 카마스피어(Karmasphere), 이클립스 플러그인(Eclipse plugin), 캅티(Cacti), 강글리아(Ganglia) 등의 도구가 만들어져 있고, 프레임워크 및 클러스터 지원을 위해 ⑫번에는 에이브로(Avro, 직렬화)와 주키퍼(Zookeeper, 코디네이터)가 있다. 더욱 향상된 고수준의 인터페이스와 활용을 위해 ⑬번의 마훛(Mahout, 기계학습), 일라스틱 맵리듀스(Elastic Map Reduce), ⑭번의 에이치베이스(HBase)로는 OLTP와 NoSQL 데이터베이스가 가능하다.

## V. 오픈소스 하둡 플랫폼의 활용

빅데이터의 활용을 위한 처리 프로세스는 6단계로 나눌 수 있다. 수집, 저장, 관리, 처리, 분석, 표현이 바로 그것이다. 물론 단계에 따라 개발자의 영역과 분석가의 영역으로 분리될 수 있으며, 다만 대용량(Volume)의 속성에 따라 병렬처리를 해야 하는 부분에서는 양자의 교차가 일어난다. 이 둘의 역할을 동시에 하는 사람을 ‘개발분석가’라고 부르며 현재 국내에서는 실제 빅데이터를 다룰 줄 아는 개발자들이 분석가에 비해 더욱 활발하게 참여하고 있다.

파일 형태이건 데이터베이스 형태이건, 또는 정형이건 비정형이건 데이터를 수집하고 통합하는 작업이 가장 우선시되어야 할 절차이다. 그 다음 이를 효율적이고 안정적으로 저장해야 하며, 저장된 데이터는 모니터링 시스템을 통해 별도로 관리되어야 한다.

빅데이터의 속성상 슈퍼컴퓨터 정도의 성능이 나오는 연산 처리가 가능해야 하며, 가설 도출 및 예측 모델 수립 등을 통한 고급 분석이 행해져야 한다. 이러한 절차를 거친 산출물은 직관적이고 명확하게 시각화되어 단순 명료한 보고서로 표현되어야 한다.

하둡 생태계에 속하는 모든 솔루션 및 도구들은 빅데이터 활

용을 위한 단계별 대응이 가능하다. 단계에 따라 가장 흔히 사용되는 대표적 도구만을 채택하여 본다면 아래와 같을 수 있다.

데이터를 수집하기 위해 Sqoop과 Flume을 사용한다. Sqoop은 RDBMS에 저장된 데이터를 하둡으로 이동시키거나 그 역과정을 수행하기 위한 것이다. Flume은 웹로그와 같은 일반 파일 형태의 데이터를 하둡에 수집하기 위한 것인데, 에이전트와 컬렉터간 통신을 통해 거의 실시간으로 데이터를 모을 수 있다.

데이터를 저장하기 위해 HDFS를 사용하고, 반정형 데이터를 저장하려 한다면 하둡 기반 NoSQL인 HBase를 별도로 설치하면 좋다. HBase는 또 다른 NoSQL 제품인 몽고DB 및 카산드라와 비교되는데 하둡을 사용할 정도의 데이터인 경우 우선적으로 권장되고 있다.

데이터 관리를 위해서는 관리 도구 및 모니터링 도구가 있다. Chukwa와 Ganglia가 대표적이며 하둡은 그 자체로 HDFS와 MapReduce의 상태를 웹페이지에서 확인할 수 있는 인터페이스를 제공하기도 한다.

연산 및 처리를 위해서는 MapReduce가 필수적이다. 하둡 클러스터를 설치하면 MapReduce도 사용이 가능한데 직접 프로그래밍하기가 쉽지 않다. 이를 위해 야후에서는 Pig를, 페이스북에서는 Hive를 개발했다. 정형 데이터의 경우 SQL 형태와 유사한 Hive를 사용하면 되고, 비정형 데이터의 경우 사용자 정의 함수 지원이 보다 원활한 Pig를 사용하면 된다.

고급분석을 위해 하둡과는 다른 오픈소스 R이 주목받고 있는 상황이다. 다만 하둡 생태계에도 Mahout이 있어 기계 학습을 통한 데이터 분석이 가능하다. 인공지능과 같이 진화하는 분석을 위해 탁월하다는 평이다.

하둡 클러스터를 구축하고 운영하고자 할 때 전체를 관장하고 조율해 주는 Zookeeper는 필수적으로 설치되어야 한다. 또한 주기적인 스케줄링을 위해서는 Oozie에게 그 역할을 맡기면 된다.

빅데이터 오픈소스 플랫폼인 하둡은 그 자체만으로는 큰 빛을 보기 어렵다. 왜냐 하면 분산 저장 및 병렬 처리에는 강할 지 모르지만 실제 서비스를 위해서는 완벽하지 않기 때문이다. 따라서 하둡과 함께 NoSQL 및 기존의 RDBMS를 병행 사용하고, 실시간 대응을 위해서는 메모리 기반의 캐시 시스템을 쓰기도 한다.

## VI. 결론

이상과 같이 빅데이터 분석 플랫폼의 표준이 되어 가고 있는 오픈소스 하둡에 대해 알아봤다. 하둡 클러스터는 기본적으로 데이터를 세 별로 복제하기 때문에 부가적인 메타 데이터를 포함하여 원본 데이터의 4배 정도의 저장 공간을 필요로 하고, 안

정적인 서비스를 위해 메모리는 최소 24Gbyte 이상, 네트워크 장비 비용은 전체 하드웨어 비용의 20% 이상을 들여야 한다. 하둡의 경우 기존의 저장 및 분석 시스템에 비해 비용이 적게 들어가는 하겠지만 장비 및 네트워크, 그리고 고급 기술자에게 들어가는 비용이 그리 적지는 않다. 빅데이터의 태생적 속성인 저렴한 비용을 만족시키기 위해 오픈소스 플랫폼 구축 및 운영에 있어 보다 다양하고 성공적인 연구 성과가 나와야 할 것이다.

## 참고 문헌

- [1] 위키백과사전, "빅 데이터", [http://ko.wikipedia.org/wiki/%EB%B9%85\\_%EB%8D%B0%EC%9D%B4%ED%84%B0](http://ko.wikipedia.org/wiki/%EB%B9%85_%EB%8D%B0%EC%9D%B4%ED%84%B0)
- [2] Sanjay Ghemawat, Howard Gobioff, Shun-Tak Leung, "The Google File System", [http://static.googleusercontent.com/external\\_content/untrusted\\_dlcp/research.google.com/ko//archive/gfs-sosp2003.pdf](http://static.googleusercontent.com/external_content/untrusted_dlcp/research.google.com/ko//archive/gfs-sosp2003.pdf), 2003
- [3] Philip Russom, "Big Data Analytics", TDWI Research Fourth Quarter, p.6., 2011
- [4] Apache Hadoop, <http://hadoop.apache.org/>
- [5] [http://creamy\\_tom.blog.me/100162288102](http://creamy_tom.blog.me/100162288102), 2012
- [6] 기가옴, "Survey, Hadoop clusters not that big not changing the world yet", <http://gigaom.com/cloud/survey-hadoop-clusters-not-that-big-not-changing-the-world-yet/>, 2012
- [7] Ken Mann, M. Tim Jones, "Distributed computing with Linux and Hadoop", <http://www.ibm.com/developerworks/linux/library/l-hadoop/>, IBM, 2012
- [8] <http://indoos.wordpress.com/2010/08/16/hadoop-ecosystem-world-map/>

## 약력



이 현 종

2001년 성균관대학교 법과대학 법학과 졸업  
 2004년~현재 Leesult 대표  
 2012년~현재 ㈜빅스터 대표이사  
 2011년~현재 빅데이터 커뮤니티 운영  
 2012년~현재 빅데이터 하둡개발자,  
 분석전문가 그룹스터디 주관  
 2012년~현재 빅데이터 오픈소스 플랫폼 기술세미나, 분석전문가 오픈세미나 주관  
 관심분야: 빅데이터 분석 및 비즈니스 활용,  
 하둡 플랫폼