**연구노트**

**(수학과)이 재 봉**  2017.06.26 (월)

|  |  |  |
| --- | --- | --- |
|  | | |
| **연 구**  **주 제** | **특허분석을 통한 미래기술 예측** | |
| **빅데이터 분석 인프라 기술 조사 – 오픈소스 중심으로** | |
| **구 분** | **연구내용** | **진행률** |
|  | 시험기간으로 6~7주차 생략  1. 하둡  [하둡이란]  하둡은 대용량 데이터를 분산 처리할 수 있는 자바 기반의 오픈 소스 프레임워크이다. 하둡은 구글이 논문으로 발표한 GFS(Google File System)과 맵리듀스(MapReduce)를 2005년 더그커팅이 구현한 결과물이다. 하둡은 분산시스템인 HDFS(Hadoop Distributed File System)에 데이터를 저장하고, 맵리듀스를 이용해 데이터를 처리한다.  하둡은 여러 대의 서버에 데이터를 저장하고, 저장된 각 서버에서 동시에 데이터를 처리하는 방식이다. 하둡은 기존의 RDBMS(Oracle,MS-SQL, MySQL등)을 대치하는 것이 아니다. 즉 트랜젝션이나 무결성을 보장해야하는 데이터처리에는 적합하지 않다. 하둡은 배치성으로 데이터를 저장하고 처리하는데 적합한 시스템이다.  쇼핑물에서 회원가입이나, 결제진행등은 모두 트랜젝션이나 무결성을 보장해야한다. 이런 것들을 하둡으로 처리하는 것이 아니라, 회원이 관심있게 보는 물품들이나, 이동경로, 머무르는 시간등 배치성으로 저장되는 데이터에 적합하다. 이런 것들을 매번 비용이 비싼 RDBMS에 저장하면 낭비요소이다. 그러므로 하둡은 RDBMS와 경쟁하는 것이 아닌 RDBMS와 협력하는 것이라 볼 수 있다.  [HDFS]  구글 분산파일 시스템을 보고 구현한 오픈소스로 기존 대용량의 파일 시스템과의 가장 큰 차이점은 저사양 서버를 이용해 스토리지를 구성할 수 있다는 점이다. HDFS에 저장하는 데이터는 물리적으로 분산된 서버의 로컬디스크에 저장되며 HDFS에서 제공하는 API를 이용해서 처리한다. HDFS는 장애복구, 스트리밍 방식의 데이터 접근, 대용량 데이터 저장, 데이터 무결성이라는 네가지 목표를 갖고 설계되었다.  [맵리듀스]  분석로직에 대해서 Map과 Reduce만 개발자가 적절히 개발하면, 데이터에 대한 분산과 병렬 처리는 프레임워크가 처리하게 된다. 네임노드에서 신규 잡이 제출되면, Job Tracker에 의해서 작업이 추적되며 이는 다시 태스크로 잘게 쪼개져서 각 데이터 노드로 전달된다. 데이터 노드에서는 입력데이터를 스플릿으로 나누어 각 맵퍼가 실행되며, 맵퍼 출력 결과는 정렬과 병합을 거쳐서 리듀서로 전달된다. 리듀서는 특별히 지정되지 않으면 1개가 동작하며 출력데이터가 다시 HDFS에 저장된다. 이러한 방식의 네임노드가 모든 잡을 추적해야 한다는 부담이 생겨서 2.0에서는 YARN을 통해 개선된 아키텍쳐가 제공된다.  [장점]  정형 데이터의 경우 기존의 관계형 데이터베이스(RDBMS)에 저장할 수 있지만 웹 로그 같은 비정형 데이터를 RDBMS에 저장하기에는 데이터 크기가 너무 크다.  상용 RDBMS가 설치되는 곳은 대부분 고가의 장비인데, 데이터를 감당하기 위해 무한정 스토리지를 늘릴 수도 없다.  이에 반해 하둡은 오픈소스 프로젝트이기에 소프트웨어 라이선스 비용에 대한 부담이 없다. 또한 값비싼 유닉스 장비를 사용하지 않고, 리눅스 서버면 얼마든지 하둡을 설치해서 운영할 수 있다. 데이터 저장 용량이 부족할 경우, 필요한 만큼 리눅스 서버만 추가 하면 된다. 하둡은 여러 대의 서버에 데이터의 복제본을 저장하고, 데이터가 저장된 각 서버에서 동시에 데이터를 처리하기 때문에  데이터의 유실이나 장애가 발생했을 때도 데이터 복구가 가능하다.  R  [R이란]  뉴질랜드 오클랜드 대학의 로버트 젠틀맨(Robert Gentleman)과 로스 이하카(Ross Ihaka)에 의해 개발되었으며, 현재 R 코어팀에 의해 지속적으로 업데이트되고 있을 뿐만 아니라, GNU GPL 라이선스 정책을 취하고 있어 전세계의 개발자에 의한 확장 패키지 개발도 활발히 이루어지고 있다. 또한 다양한 운영체제를 지원하여 통계학자나 연구자들을 비롯한 많은 일반 사용자들에게도 통계 계산, 행렬 처리 등을 위한 도구로 널리 사용되고 있다.  R은 존 챔버스(John Chambers)가 개발한 S언어 기반으로 개발되었으며, 데이터 처리는 LISP언어의 일종인 스킴 언어(Scheme)의 영향을 받아 함수형 프로그래밍과 절차적 프로그래밍의 성격을 갖고 있다. 그러나 Java, Python, C++ 등의 다양한 언어와 호환할 수 있는 라이브러리 등이 개발되어 있으며, 서버 형태로 운용할 수 있는 등 필요에 따라 유연하게 활용할 수 있다.  [장점]  R의 최대 장점은 앞서 설명한 바와 같이 무료 오픈소스라는 특징으로 인해 수많은 라이브러리가 개발되어 공유된다는 점이다. 이러한 특징으로 인해 통계처리 뿐만 아니라 자연어 처리, 기계학습, 시맨틱 기술 등 까지도 쉽게 접목할 수 있다. 구현한 결과를 그래프 등으로 시각화 할 수 있으며, JAVA나 C, Python 등의 다른 프로그래밍 언어와 연결도 용이하다. 게다가 MAC, OS, 리눅스/유닉스, 윈도우 등의 대부분의 컴퓨팅 환경을 지원하는 것도 장점이다. 위의 장점들로 인해 R은 통계분석 분야에서 인지도를 높여왔으며, 하둡 환경 상에서 분산처리를 지원하는 라이브러리 덕분에 구글, 페이스북, 아마존 등의 빅데이터 분석이 필요한 기업에서 대용량 데이터 통계분석 및 데이터 마이닝을 위해 널리 사용되고 있다.  NoSQL  [NoSQL이란]  NoSQL은  Not Only SQL의 약자로 기존RDBMS 형태의 관계형 데이터베이스가 아닌 다른 형태의 데이터 저장기술을 의미한다. NoSQL이라고해서 RDBMS제품군(MS-SQL, Oracle, Sybase, MySQL) 등과 같이 공통된 형태의 데이터 저장방식(테이블)과 접근방식(SQL)을 갖는 제품군이 아니라 RDBMS와 다른 형태의 데이터 저장구조를 총칭하며, 제품에 따라 각기 그 특성이 매우 달라서 NoSQL을 하나의 제품군으로 정의할 수 없다.  [특징]  가장 큰 특징 중에 하나는 관계형 데이터베이스인 RDBMS가 데이터의 관계를 Foreign Key 등으로 정의하고 이를 이용해 Join등의 관계형 연산을 한다고 하면, NoSQL은 데이터 간의 관계를 정의하지 않는다. 데이터 테이블은 그냥 하나의 테이블이며 각 테이블 간의 관계를 정의하지도 않고 일반적으로 테이블 간의 Join도 불가능하다. 또한 RDBMS의 복잡도와 용량 한계를 극복하기 위한 목적으로 등장한 만큼, 페타바이트급의 대용량 데이터를 저장할 수 있다.  NoSQL은 기존의 RDBMS와는 다르게 하나의 고성능 머신에 데이터를 저장하는 것이 아니라, 일반적인 서버(인텔 계열의 CPU를 사용하는 Commodity Server) 수십 대를 연결해 데이터를 저장 및 처리하는 구조를 갖는다. 즉 분산형 구조를 통해 데이터를 여러 대의 서버에 분산해 저장하고, 분산 시에 데이터를 상호 복제해 특정 서버에 장애가 발생했을 때에도 데이터 유실이나 서비스 중지가 없는 형태의 구조이다. |  |
| **차 주**  **계 획** | ■ 계획  - 관련 논문 조사 | |
| **이 슈** | ■ **해당사항 없음** | |

|  |  |
| --- | --- |
|  |  |