

미래 인재로의 도약, 스마트기술 기반 다지기	
07차시	빅데이터 플랫폼과 관련 기술

1. 빅데이터 순환과정과 플랫폼

1) 빅데이터 순환과정

빅데이터를 분석하여 의미 있는 지식을 찾아 활용하기 위해서는 여러 단계를 포함하는 순환 과정을 거치게 된다. 먼저 발굴하고자 하는 지식과 관련된 다양한 데이터 소스(data source)로부터 데이터를 수집하고, 수집한 데이터에서 필요 없는 데이터를 필터링하거나 적절한 형태로 가공하는 등 전처리 단계를 거친다. 그 후에 데이터를 체계적으로 저장하고 관리하면서 데이터 처리 과정을 통해 의미 있고 유용한 지식을 발굴하고 시각화하게 된다.

빅데이터 순환 과정에서 데이터 소스나 지식을 활용하는 서비스 분야가 무엇인지에 따라 일부 단계를 건너뛰거나 반복 수행되기도 한다. 이처럼 빅데이터는 지식을 최종 활용하기까지 데이터 수집, 저장 관리, 처리·분석 및 데이터(지식) 시각화, 이용, 폐기 등을 통한 여러 단계를 거치고 있다.

2) 빅데이터 플랫폼

다양한 데이터 소스에서 수집한 데이터를 분석 처리하여 지식을 추출하고, 이를 기반으로 지능화된 서비스를 제공하는 데 필요한 ICT 환경을 빅데이터 플랫폼이라 한다. 빅데이터 플랫폼이 빅데이터 처리에 필요한 전체의 순환 과정을 수행하기 위해서는 확장성 있는 대용량 처리 능력, 이기종 데이터 수집 및 통합 처리 능력, 빠른 데이터 접근 및 처리 능력, 대량의 데이터를 저장 관리 할 수 있는 능력, 대량의 이기종 데이터를 원하는 수준으로 분석할 수 있는 능력 등을 갖춰야 한다. 다양한 데이터 소스로부터 데이터 수집, 저장 관리, 처리·분석 및 지식 시각화를 통해 지식을 이용하기까지 각 단계를 지원하는 데 필요한 공통 소프트웨어를 빅데이터 플랫폼(Big data analytics platform)이라고 한다.

이러한 빅데이터 플랫폼은 빅데이터 수집 기술, 빅데이터 저장/관리 기술, 빅데이터 처리 기술, 빅데이터 분석 기술 및 지식 시각화 기술 등을 적용하여 구현한다. 빅데이터로부터 지식을 얻어 활용하기까지는 여러 단계가 필요하고, 그 단계마다 수많은 기술이 활용된다. 빅데이터 플랫폼은 데이터를 수집해서 지식을 발굴하는 데 필요한 빅데이터 처리 플랫폼 기술과 대용량의 고속 저장 공간 및 고성능의 계산 능력을 갖춘 컴퓨터 등 컴퓨터 기반을 제공하는 빅데이터 컴퓨팅 인프라 기술로 구성된다.

3) 데이터 관리 개념과 유형

가. 데이터 관리와 데이터베이스

데이터란 숫자, 영상, 단어 형태의 단위를 뜻한다. Datum(데이터의 단위)이 여러 개 모여 데이터(data)가 형성하며, 다양한 데이터가 의미 있는 가치를 형성하면 정보 (information)가 된다. 데이터베이스(DB)는 데이터의 효율적이고 통합적인 저장을 가리키는 용어이다. 서로 관련되는 데이터를 효율적으로 관리하기 위해 수집한 데이터의 집합체이며, 각 데이터는 상호 유기적 관계에 의해 구성되어 있다. 데이터 관리는 전통적으로 파일 관리에서부터 시작하여 그 편리성을 도모하는 방향에서 데이터베이스로 발전되어 왔고 여러 가지 진화하는 과정을 거쳐 왔다.

먼저 파일을 관리하는 파일 시스템(file system)은 컴퓨터 파일에 이름을 붙이고, 저장 및 검색을 위해 논리적으로 그것들을 어디에 위치시켜야 하는지 등을 나타내는 방법이다. 파일은 실제 데이터와 메타 데이터(파일의 위치, 크기, 소유자, 허가권 등)를 유지·저장하는 체계로 구성된다.

그동안 파일은 데이터 관리와 연관하여 여러 문제들을 야기시켰는데, 이러한 문제들을 해결하기 위하여 제시된 대안이 데이터베이스이다. 1960년대 말에 처음으로 '데이터베이스(DB, database)'란 용어가 생성되어 '한 조직의 응용 시스템들을 공유(shared)하기 위해 통합(integrated), 저장(stored)한 운영(operational) 데이터의 집합'이란 개념으로 정의되었으며, 이어서 데이터베이스를 관리하기 위한 시스템인 DBMS(Database Management System)가 탄생하였다.

데이터베이스(database)란 상호 관계가 있는 자료들을 하나의 통합된 저장소에 논리적으로 저장한 것을 의미한다. 데이터베이스는 파일 시스템과는 달리 조직 내 다수의 사용자나 다수의 부서 또는 조직 전체의 정보 요구 사항에 부응할 수 있는 자료의 공유와 통합 관리를 목표로 한다. 여러 부서에서 따로 보관하고 있던 여러 개의 자료가 하나만 존재하여도 되므로 자료 중복을 배제할 수 있고, 한 곳에 집중되어 있는 자료를 모든 부서의 사용자가 쉽게 공유할 수 있는 체계를 갖추고 있다. 파일 시스템이나 데이터베이스 시스템은 문맥에 따라 시스템 자체를 뜻하기도 하며, 이들을 이용한 데이터 관리 방법을 의미하기도 한다.

또한 전사적인 수준에서 기업의 의사결정에 필요한 관련 데이터를 과거의 데이터까지 포함하여 축적한 별도 저장소인 대용량의 데이터웨어하우스(data warehouse), 고객관리, 상품관리, 재무/회계 등의 단일 주제별 또는 지역별, 단일 부서 또는 사용자 집단 등 의사결정 그룹별로 구축되는 데이터 마트(data mart)도 넓은 의미에서 데이터베이스에 포함된다.

나. 빅데이터 관리 방법(품질 기준)

빅데이터는 대량의 데이터, 미세하고 정밀한 데이터 및 데이터 소유자 불분명 등의 특성에 따라 전통적인 데이터베이스와 다르게 관리 방법과 체계를 갖추어야 한다. 이를 정리하면 다음과 같다.

첫째, 빅데이터는 통상적으로 수작업으로 수집되기보다는 기계, 프로그램 등에 의해 수집되는 대량의 데이터라는 관점에서 혹시 발생할지 모르는 데이터 사용자의 오류는 무시할 수 있다. 따라서 데이터 수집 과정의 타당성을 방해하는 예외 상황을 탐지하는 수준으로 품질

기준을 정의하여야 한다.

둘째, 인터넷 사용자가 웹 서핑을 하는 동안 방문한 사이트 목록인 클릭 스트림(click stream), 미터 값 등 기계, 센서, 프로그램 등에서 생산되는 데이터이므로 기존 데이터보다 훨씬 미세한 데이터라는 점에서 개별 데이터에 대한 타당성 검증은 경우에 따라 불필요하다. 결국 개별 레코드에 대한 의미보다 데이터 전체가 나타내는 의미를 중심으로 품질 기준을 정의해야 한다.

셋째, 빅데이터는 누가 언제 어디서 데이터를 생산한 것인지에 대한 관리·감독이 불가능한 조직 외부의 데이터이다. 그러므로 데이터 소유자가 불분명한 만큼 목적이나 통제 없이 생산된 데이터에 대한 데이터 품질 기준을 정의하기 위한 다른 방법 모색이 필요하다. 한편, 빅데이터는 기존 정형화된 데이터뿐만 아니라 비정형화된 데이터를 포함하고 그 규모 또한 방대하여 기존의 데이터 베이스로는 저장 및 관리에 한계가 있어 빅데이터를 저장하고 관리하기 위한 새로운 형태의 저장 및 관리 기술·방법을 채택하여야 한다.

2. 빅데이터 수집 및 저장 기술

1) 빅데이터 수집 방법

가. 빅데이터 수집

빅데이터 수집은 분산된 다양한 데이터 소스(source)로부터 필요로 하는 데이터를 수동 혹은 자동으로 이루어진다. 따라서 조직 내부에 분산된 정형 데이터의 수집과 조직 외부에 흩어진 비정형 데이터의 수집을 모두 고려하여야 하며, 이는 통상 다양한 도구와 프로그래밍 등에 의해 자동으로 이루어진다. 여기에는 로그 수집기, 크롤링(crawling), 센싱(sensing) 등 다양한 방법이 있다. 빅데이터의 자동 수집 방법은 로그 수집기, 크롤링, 센싱, RSS 리더 등 4가지가 있다.

먼저, 로그 수집기에는 다양한 오픈 소스 솔루션이 있는데, 대표적인 것으로 스크라이브(Scribe), 플럼(Flume), 척화(Chukwa) 등이 있다. 이들은 끊임없이 들어오는 빅데이터를 연속적으로 처리하는 것에 적합한 도구들이다. 이 대표적인 오픈 소스 솔루션들에 대한 특징을 간략히 정리하면 표와 같다.

구분	설명
스크라이브(Scribe)	<ul style="list-style-type: none">- 분산된 서버에서 발생하는 데이터를 중앙 집중 서버로 전송하는 방식으로 로그 저장을 위해 다양한 저장소를 활용 가능- 설치 및 구성이 용이
플럼(Flume)	<ul style="list-style-type: none">- 커다란 규모의 분산 데이터를 수집하고 효율적으로 전송하는 시스템- 클러스터 환경에서 신뢰성 있는 로깅 뿐만 아니라 안정적인 확장성을 제공- 다양한 정비로부터 수집되고 모아지는 데이터를 하둡 같은 중앙 처리 저장 시스템에 저장해주는 역할

	- 주된 설계 목적은 신뢰성, 가용성, 관리성, 그리고 확장성
척화(Chukwa)	<ul style="list-style-type: none"> - 분산되어 있는 노드들의 다양한 로그를 수집하여 HDFS(Hadoop Distributed File System)에 저장하고 프로세싱하는 시스템 - 사용자는 HICC(Hadoop Infrastructure Care Center)라는 웹 포털 인터페이스를 통해서 초 단위로 생성되는 파일이나 블록수와 같은 HDFS의 상태를 실시간으로 모니터링 가능 - 중복 제거는 맵리듀스 작업으로 처리

다음으로 크롤링은 웹페이지에 게시되어 있는 정보를 수집하는 기법으로 웹 로봇과 웹 크롤러 등을 사용한다. 각 기법의 특징은 표에 제시된 바와 같다.

구분	크롤링 기법의 특징
웹 로봇 (Web Robot)	<ul style="list-style-type: none"> - 지정된 URL 리스트에서 시작하여 웹 문서를 수집하고, 수집된 웹 문서에 포함된 URL 들을 추출하여 새롭게 발견된 URL에 대한 웹 문서 수집 과정을 반복하는 소프트웨어 - 웹 문서를 돌아다니면서 필요한 정보를 수집하고 이를 색인해 정리하는 기능을 수행하며 주로 검색엔진에서 사용되고 있음 - 일반적으로 수집기와 분류기, 데이터 처리기로 구성
웹 크롤러 (Web Crawler)	<ul style="list-style-type: none"> - 조직적, 자동화된 방법으로 월드와이드웹(World Wide Web)을 탐색하는 컴퓨터 프로그램 - 방문한 사이트의 모든 페이지 복사본을 생성하는데 사용되며, 검색엔진은 이렇게 생성된 페이지의 보다 빠른 검색을 위하여 인덱싱을 수행 - 링크 체크나 HTML 코드 검증과 같은 웹사이트의 자동 유지 관리 작업을 위해 사용되기도 하며, 자동 이메일 수집과 같은 웹페이지의 특정 형태의 정보를 수집하는 데에도 사용

나. 수집된 빅데이터의 변환

데이터 전처리 단계에서 수집된 데이터를 전환하는 기술은 데이터 수집·통합 단계에서 수집된 데이터 중 비정형 데이터를 정제하여 분석 가능한 형태로 구조화하여 분석의 정확성을 높이고 심층 분석을 가능하게 하는 기술을 말한다. 즉 데이터 통합, 익명화, 정제, 검색, 인증 및 ETL(Extraction, Transformation, Loading) 등의 데이터 변환 기술을 통해 다음 단계로의 진입을 원활하게 해야 한다.

정보 검색 기술의 목표는 내가 원하는 정보를 신뢰성 있고, 편안하고, 빠르게 검색하는 것이다. 따라서 빅데이터의 수집은 수집 기능 외에도 데이터를 저장하거나 분석하기 위해 데이터를 변환하거나 통합하는 작업도 포함하고 있다. 데이터의 변환 및 통합은 레거시 데이터 간 통합, 비정형 데이터의 정형화, 레거시 데이터와 비정형 데이터 간의 통합 측면 등에 대해 고려해야 한다. 특히, 비정형 데이터는 데이터를 효과적으로 분석하기 위해 데이터를 수집하는 과정에서 구조적 형태로 전환되어 저장되어야 하며, 레거시 데이터(legacy data)와 수집한 비정형 데이터 간의 통합 방안도 고려되어야 한다. 기업 내 운영 환경에서 특정 데

데이터베이스 시스템에서 발생하거나 변경된 데이터를 다른 시스템에 적용하려는 분산 및 복제 환경은 보편화되어 있다.

마찬가지로, 서로 다른 시스템 간의 데이터 공유는 빅데이터 시스템에 있어서도 중요하고 필수적인 이슈가 된다. 데이터 공유를 위한 가장 일반적인 형태는 운영계 시스템의 데이터 복제(replication) 기술과 정보계 시스템을 위한 데이터웨어하우스의 ETL(Extract, Transformation, Load; 추출, 변환, 로드) 프로세스이다. ETL 프로세스는 필요한 데이터 소스로부터 데이터를 추출하여 분석 가능한 데이터로 변환하는 것을 가능하게 한다.

다. 빅데이터 수집 시스템

빅데이터 수집 시스템은 다양한 데이터 소스로부터 필요한 데이터를 수집하기 위해 확장성, 안정성, 실시간성, 유연성 등을 확보하고 있어야 한다.

2) 빅데이터 저장 방법

가. 빅데이터 저장 기술

데이터 저장 기술은 웹 데이터, 소셜 미디어, 비즈니스 데이터, 센싱 데이터 등 날로 증가하는 다양한 형식의 데이터를 실시간으로 저장·관리할 수 있는 분산 컴퓨팅 기술과 관련이 있는 것으로서 빅데이터 플랫폼의 핵심 기술 중 하나이다. 이러한 빅데이터 저장 기술은 구글이나 애플 등에 의해 지속적으로 진화하고 있으며, 대표적인 솔루션으로 하둡의 HDFS/Hbase, Cassandra, MongoDB 등이 있다. 우리나라에서 개발한 한국전자통신연구원(ETRI)의 GLORY-FS도 이에 해당한다.

대용량 데이터를 저장하기 위한 관련 기술은 분산파일시스템, NoSQL, 병렬 DBMS, 네트워크 구성 및 클라우드 파일저장시스템 등이 있다. 분산파일시스템의 대표격인 하둡은 저렴한 비용으로 빅데이터 시스템을 구축할 수 있어 가장 널리 사용되고 있다. NoSQL과 병렬 DBMS는 수평 확장 접근 방식을 사용하는 기술이다. 이외에도 서로 다른 종류의 데이터 저장 장치를 한 데이터 서버에 연결하여 총괄적으로 관리해주는 네트워크인 SAN(Storage Area Network)이나 컴퓨터 네트워크가 연결된 곳에서는 언제 어디에서라도 스토리지에 접근해서 사용할 수 있도록 하는 저장 장치인 NAS(Network Attached Storage)와 같은 네트워크 구성 저장 기술, Amazon S3나 OpenStack Swift와 같은 클라우드 파일저장시스템 등이 있다.

특히 NoSQL(Non- SQL 또는 Not-only-SQL)은 기존의 오라클과 같은 관계형 데이터베이스(RDBMS) 중심의 데이터 저장 기술에 의해서는 저장 및 관리가 어려운 비정형 데이터에 관한 새로운 데이터 저장 기술이다. 이러한 NoSQL은 데이터를 저장할 때 키값(key-values)과 같은 간단한 구조로 저장을 하고 데이터를 저장하거나 읽을 때 SQL(Structured Query Language, 구조적 질의 언어)을 사용하지 않는다. 따라서 전통적인 RDBMS와 다르게 SQL 없이 데이터를 관리하는 비관계형 데이터베이스라 할 수 있다. 관계형 데이터베이스(RDBMS)에서 SQL은 여러 테이블을 조인할 때 최적의 경로를 선택하는 역할을 하는데, NoSQL은 한 테이블에 대해 데이터를 넣거나 질의함으로써 굳이 SQL을 사용할 필요가 없는

것이다. 대표적인 NoSQL 솔루션으로는 Cassandra, Hbase, MongoDB 등이 있다.

나. 빅데이터 저장을 위한 고려 요소

빅데이터의 저장과 관련하여 저장 단가를 절감할 수 있는 비용 문제, 자료 저장과 인출 속도를 향상시킬 수 있는 성능의 문제, 저장의 신뢰도와 안정성을 보장하는 문제, 저장 공간의 확장성 등은 핵심적인 고려 요소이다. 또한 기존의 시스템과 빅데이터 저장 시스템 전반에 걸쳐 콘텐츠 인덱스 유지·관리, 데이터 스키마 및 구조에 상관없이 검색, 수집, 저장, 편집할 수 있는 환경 구축 등의 필요하다. 빅데이터의 저장 시 고려해야 할 문제점 및 기술 요건에 대한 내용을 정리하면, 표와 같다.

현실적 문제점	기술 요구 사항
<ul style="list-style-type: none"> - 데이터 저장과 관리에 고비용 소요 - 저장된 데이터의 효율적 관리 어려움 - 용량한계 봉착 시 확장 곤란(기술, 비용) 	<ul style="list-style-type: none"> - 데이터 저장에 낮은 총소유 비용(Total cost of ownership, TCO) 실현 - 대용량 데이터 저장 - 수평적 확장 용이성

3. 빅데이터 처리 및 인프라 기술

1) 빅데이터 처리 기술의 개요

빅데이터의 처리 단계는 고객관계관리(CRM)에서 행해지는 수집 분석의 흐름과 비슷하다. 웹, SNS 로그, RDBMS 등을 통해서 수집된 대용량 데이터를 저장하고, 분산·병렬 데이터 분석 기법을 이용하여 빅데이터를 분석하고, 다양한 분야에서 데이터를 활용하는 단계로 이루어지며 이때 빅데이터 요소 기술이 투입된다.

빅데이터의 처리 관련 기술로 지속적인 연구 개발을 통해 새로운 요소 기술이 등장하고 있다. 대용량 데이터 처리 능력을 위한 분산처리 기술로는 하둡 분산파일시스템(HADOOP Distributed File System), H베이스(HBase), 맵리듀스 등이 있다. 그리고 메모리상에 필요한 데이터와 인덱스(index)를 보관함으로써 데이터 검색 시간을 크게 줄일 수 있는 인메모리(In-Memory) 기술도 개발되었다.

하둡(HADOOP)은 현재 빅데이터 처리에서 가장 선호되는 솔루션으로서 HDFS에 파일을 저장하고, 맵리듀스 방식으로 많은 양의 데이터를 빠른 속도로 분산 병렬 처리한다. 현재 포춘 500대 기업 모두가 하둡을 활용하고 있으며, 2022년 하둡 시장은 100조 원 규모로 커질 것이라는 전망이다(Zion Market Research, 2017). 저가의 상용 하드웨어 환경에서 확장성이 뛰어나 수많은 하위 프로젝트(sub-project)들과 하둡 생태계시스템(Hadoop ecosystem)을 형성하면서 초기에는 HDFS(저장), MapReduce(처리) 프레임워크로 시작되었으나, 데이터 저장, 실행 엔진, 프로그래밍 및 데이터 처리와 같은 하둡 생태계 전반을 포함하는 의미로 확장 발전하고 있다.

2) 분산 컴퓨팅과 하둡

가. 분산 컴퓨팅의 이해

분산 컴퓨팅(distributed computing)이란 여러 대의 컴퓨터를 연결하여 상호 작용하게 함으로써 컴퓨팅의 성능과 효율을 높이는 것을 말한다. 넓은 의미의 분산 컴퓨팅은 여러 개의 컴퓨팅 자원을 하나의 시스템 안에 결합하여 연결한 병렬 컴퓨팅까지 포함시키기도 한다. 분산 컴퓨팅의 목적은 성능 확대(scalability)와 가용성(high availability) 확대에 있다. 성능 확대를 위한 분산 컴퓨팅의 대표적인 예로써 컴퓨터 클러스터(computer cluster)의 활용인데 수직적 성능 확대(vertical scalability)와 수평적 성능 확대(horizontal scalability)가 있다.

먼저 수직적 성능 확대(vertical scalability)와는 컴퓨터 자체의 성능을 업그레이드하는 것을 말한다. 즉, 연산(operation) 능력이 강한 CPU의 사용, 주기억 장치와 하드디스크 등의 증설로 인한 성능 향상을 의미한다. 하지만 통신 속도로 인한 지연을 해결하기 위해 기가급(G-bps) 이상의 통신 연결을 통해 문제점을 해결해야 한다. 입·출력(input/output) 측면에서는 Non-blocking I/O와 비동기적 입·출력을 통해 효율 증대와 멀티스레딩(multi-threading) 등을 통해 해결해야 한다. 결과적으로 수직적 성능 확대는 기존의 프로그램 등 각종의 운영 환경의 변화 없이 업무를 지속할 수 있다는 장점이 있는 반면, 컴퓨터가 고가 사양이 될수록 비용이 커진다는 단점이 있다.

한편, 수평적 성능 확대(horizontal scalability)는 컴퓨터들을 네트워크로 연결하여 성능을 업그레이드하는 것을 말한다. 단순히 컴퓨팅 노드(node)의 수를 늘리고 상호 연동하여 사용해서 성능을 향상하는 방법이다. 또한 기능상 Peer-to-Peer 모델과 Master-Slave 모델 등이 있다. 이는 기존에 고가의 성능이 뛰어난 큰 컴퓨터를 사용하거나 업그레이드하는 방법과 반대되는 말이다. 이 경우 무엇보다 기존 투자 지원을 이용할 수 있고, 점진적인 성능 개선을 추구할 수 있다는 장점이 있다.

클라우드 컴퓨팅 환경을 구축하기 위한 다양한 분산파일시스템이 존재한다. 대표적으로 알려진 것은 구글의 파일 시스템인 GFS와 아파치 프로젝트의 일부분인 하둡의 HDFS이다. GFS는 구글의 기술로 비공개되어 있어 다른 사용자는 사용할 수 없다. 따라서 상용 서비스를 하고 있는 공공 클라우드 컴퓨팅 서비스 제공자는 하둡의 HDFS를 사용하고 있다. 왜냐하면 하둡은 오픈 소프트웨어로서 사용상 제약이 없고 서비스 제공자의 서비스 환경에 따라 다양하게 수정하여 사용할 수 있어 많은 서비스 제공자가 사용 중이다.

나. 하둡

빅데이터 환경에서 생산되는 데이터는 그 규모와 크기가 방대하기 때문에 기존의 파일 시스템 체계를 그대로 사용할 경우 많은 시간과 높은 처리 비용을 필요로 한다. 따라서 대용량의 데이터를 분석하기 위해 두 대 이상의 컴퓨터를 이용하여 적절히 작업을 분배하고 다시 조합하며, 일부 작업에 문제가 생겼을 경우 문제가 발생한 부분만 재처리가 가능한 분산 컴퓨팅 환경을 요구한다. 이를 지원하는 가장 대표적이며 널리 알려진 도구가 하둡(Hadoop)이다.

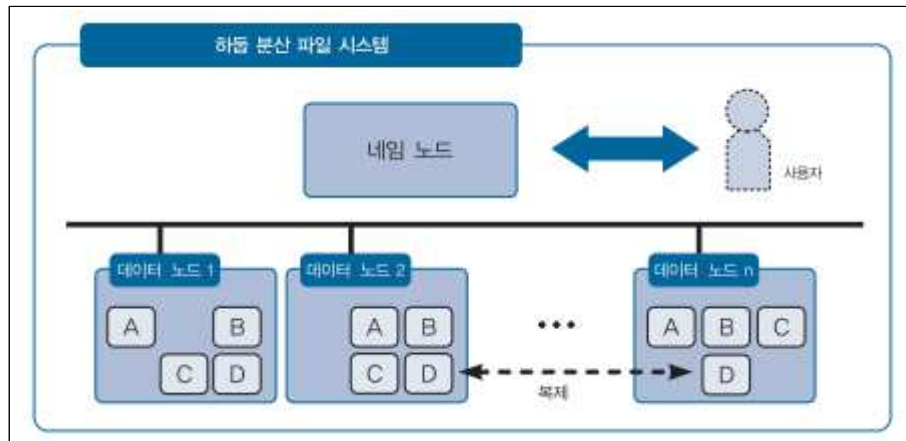
하둡은 대용량의 데이터를 처리하기 위해 대규모의 컴퓨터 클러스터에서 동작하는 분산 애플리케이션 개발을 위한 자바 오픈소스 프레임워크이다. 하둡은 2005년 더그 커팅과 마이크 파렐라가 공동 개발한 파일 시스템으로, 구글의 분산 파일 시스템(GFS, Google File System)에 대응하기 위해 만들어졌다. 하둡은 검색 엔진의 웹문서 검색 기술에 착안하여, 수 천 대의 파일 시스템에 데이터를 분산하여 저장하고 필요 시 각 시스템의 CPU와 메모리 자원을 병렬적으로 활용하여 빠른 데이터 처리를 가능케 한 데이터 처리 기술이다. 특히 하둡은 초기 구글의 병렬 분산 컴퓨팅 시스템으로 개발된 맵리듀스(MapReduce)와 결합할 때 최대의 효율을 내는 것으로 알려져 있다.

분산파일시스템의 발달은 비정형 데이터의 처리 방식에도 영향을 주게 되었다. 하둡은 웹 검색을 기반으로 하여 발달된 기술이므로, 정제된 데이터베이스를 필연적인 전제 조건으로 삼지 않는다. 즉, 사용자가 원하는 데이터를 찾고자 할 때에는 분산된 데이터 소스로부터 곧바로 정보를 추출하는 것이 가능하게 해 준다. 이에 따라 정제된 데이터베이스를 기반으로 하는 것이 정보시스템 구성에 반드시 필요하지 않게 되었다. 따라서 과거 관계형 데이터베이스관리시스템(RDBMS)과 밀접하게 연계되었던 SQL을 활용한 정보 추출 방식의 중요성이 감소하게 되었고, 반면에 분산 데이터베이스에서 정보를 얻을 수 있는 새로운 정보 처리 기술이 필요하게 되었다. 비정형 데이터 분석에는 주로 웹 문서 검색에 적용된 기법들이 사용되고 있다. 최근에는 이미지 정보로부터 직접 패턴을 추출하여 의미를 분석해내는 기법도 시도되고 있으나, 아직까지는 텍스트를 중심으로 의미를 찾아내는 분석에 집중되고 있다.

하둡은 대용량 데이터의 분산 저장 및 신속한 처리를 위해 다수의 컴퓨터를 네트워크로 연결하여 하나의 시스템과 같이 사용할 수 있도록 구성한 시스템이다. 하둡은 신뢰할 수 있고 (reliable), 확장이 용이하며(scalable), 분산 컴퓨팅 환경을 지원하는 오픈소스 소프트웨어이다. 최근 야후, 아마존닷컴 등 많은 기업에서 빅데이터의 처리를 위해 하둡을 활용하고 있으며, 사실상 하둡은 빅데이터의 처리를 위한 표준 플랫폼으로 인식되고 있다.

하둡의 HDFS는 클라우드 컴퓨팅 환경을 구축하기 위해 이용하게 된다. 분산 파일 시스템은 일반적으로 데이터의 손실을 예방하기 위해 많은 저장 장치를 사용하며 사용자에게 의해 저장되는 데이터를 안전하게 보관해야 하고, 수시로 발생하는 하드웨어 장애에 서비스가 중단되지 않도록 대비해야 한다. 또한 데이터의 입·출력 처리를 원활히 수행할 수 있는 성능을 가지고 있어야 한다.

하둡 시스템은 마스터 노드(master node)와 슬레이브 노드(slave node)들을 하나의 클러스터로 묶어 이루어져 있으며, 기능적 측면으로는 크게 HDFS와 맵리듀스 시스템으로 구성되어 있다. 특히 HDFS는 대용량 데이터의 분산 저장 기능을 제공하는 시스템인데, 하둡 클러스터에 있는 마스터 노드 중 하나를 네임 노드(name node), 그리고 슬레이브 노드들을 데이터 노드(data node)라고 부르고 있다.



그림과 같이 HDFS에서 데이터를 저장하면, 데이터는 블록 단위로 나누어 HDFS를 구성하고 있는 데이터 노드에 자동으로 분산 저장된다. 블록이 분산되어 저장되면, 해당 블록을 저장한 데이터 노드는 그 블록을 2개의 다른 데이터 노드로 전송하여 같은 블록 3개를 항상 유지할 수 있게 한다. 블록의 수를 3배수로 저장하고 있어 데이터의 가용성을 높인다. 데이터 노드들은 3초 주기로 서로 메시지를 주고받으며 데이터 노드의 상황을 체크한다.

HDFS는 고장 감내(fault-tolerant)를 위해 고비용의 하드웨어를 사용하는 기존의 서버 구성 방식과 다르게 리눅스를 운영 체제로 하는 저비용 하드웨어를 통해 구축이 가능하도록 설계되어 있다. 또한 자바 언어를 이용하여 다양한 서버에서 구동할 수 있다. HDFS는 파일 생성, 삭제, 이동, 수정 등이 가능하지만 사용자의 직접적인 접근 권한과 링크는 지원하지 않는다. 데이터 노드가 사용할 수 없는 상태라고 판단되면 해당 블록과 같은 것을 가지고 있는 데이터 노드는 다른 데이터 노드에게 전송하여 3개의 블록을 유지하도록 한다. 데이터 노드의 결함으로 인해서 손상된 블록은 데이터 노드가 정상화되면 같은 블록을 가지고 있는 데이터 노드에 의해서 다시 재전송되어 정상화 작업을 실행한다. 블록의 복구를 위해 재전송되는 복사 과정은 블록의 가용성을 향상시키는 장점을 가지게 하지만, 데이터의 수정으로 인해 업데이트가 발생할 경우 원본 데이터의 블록인지 확인할 방법이 없다.

다. 맵리듀스

맵리듀스(MapReduce)는 컴퓨터 클러스터 환경에서 페타 바이트(Petabyte) 이상의 대용량 데이터를 병렬 처리하기 위해서 만들어진 프로그램 모델이다. 맵리듀스는 리스트 형식의 데이터를 처리할 때 사용하는 함수인 맵(map) 함수와 맵 작업에서의 결과를 취합하는 리듀스(reduce) 함수를 기반으로 구성되어 있다. 맵리듀스는 사용자가 정의한 프로그램을 작업(job)이라는 단위로 관리하며, 작업은 여러 대의 노드에서 분산처리 되는데, 이때 각 노드에서 수행되는 세부 작업을 태스크(task)라 부른다. 태스크의 처리 과정은 데이터를 여러 개의 데이터 조각으로 나눠서 분산 처리한 후 필요에 따라 그 결과를 하나로 모아 처리하고, 최종 결과를 다시 분산 파일 시스템에 저장하는 방식이다. 이때 분산 처리하는 과정을 맵 태스크라고 하고, 각각의 결과를 하나로 모아 처리하는 과정을 리듀스 태스크라 한다.

3) 실시간 처리 기술

가. 맵리듀스의 배치처리 방식

빅데이터를 처리하는 프레임워크로 대부분 맵리듀스를 사용한다. 맵리듀스는 페타 바이트 이상의 대규모 데이터를 여러 노드로 구성된 클라우드 환경에서 병렬 처리하는 기법으로, 함수형 프로그래밍에서 일반적으로 사용되는 맵과 리듀스 방식을 사용해 데이터를 처리한다. 맵리듀스는 대량 데이터를 분산 처리할 수 있는 기법이지만, 일괄 처리(batch processing) 방식으로 데이터를 처리하기 때문에 실시간으로 데이터를 조회하기는 어렵다.

나. 실시간 처리 기술

이런 단점을 극복하기 위해 최근 몇 년간 실시간 분산 쿼리나 스트리밍 처리 기법이 많이 연구되었다.

실시간 분산 쿼리(Realtime Distributed Query)는 클러스터를 구성하는 노드가 각자 쿼리를 처리하고 한 번에 처리할 데이터의 크기는 작게 하면서 이를 병렬 처리해 응답 시간을 실시간 수준으로 높이는 방식이다. 스트리밍 처리(Streaming Processing)는 끊임없이 들어오는 데이터를 유입 시점에 분석해 원하는 데이터 뷰로 미리 만드는 방식이다. 이 방식은 CEP(Complex Event Processing)라고도 부르며, 트위터의 스톰(storm)과 아파치 스파크(spark)가 이 방식에 속한다.

아파치 스파크(Spark)는 2009년 버클리대학교의 AMPLab에서 소소하게 시작된 이후, 현재 발전을 거듭해서 세계에서 가장 중요한 빅데이터 분산 처리 프레임워크 가운데 하나로 부상하였다. 스파크는 다음과 같은 장점이 있다. 첫째, 전 세계에서 가장 많이 사용되고 있는 분산 처리 플랫폼이다. 둘째, 다량의 데이터를 다량의 서버에서 동시에 분석할 수 있는 플랫폼을 제공한다. 셋째, 기존의 SQL을 활용하여 분석 결과를 가져올 수 있다. 넷째, 데이터 전처리를 위해 파이썬(Python)과 같은 언어를 사용할 수 있다.

또한, 스파크는 다양한 방법으로 배포가 가능하다. 자바, 파이썬, R 등의 프로그래밍 언어를 위한 네이티브 바인딩을 제공하고 있으며 SQL, 스트리밍 데이터, 머신러닝, 그래프 처리 등을 지원한다. 이에 전 세계의 주요 디지털 기술 기업들이 아파치 스파크를 사용하고 있다.