

목차

- Hadoop ECO System 2.0 3
 - 코디네이터 3
 - 리소스 관리 3
 - 데이터 저장 4
 - 데이터 수집 4
 - 데이터 처리 5
 - 워크플로우 관리 6
 - 데이터 시각화 6
 - 데이터 직렬화 6
 - 참고 6

Hadoop ECO System 2.0

출처 : <http://blrunner.com/99>



코디네이터

- Zookeeper(<http://zookeeper.apache.org>) 분산 환경에서 서버 간의 상호 조정이 필요한 다양한 서비스를 제공하는 시스템으로, 크게 다음과 같은 네 가지 역할을 수행합니다. 첫째, 하나의 서버에만 서비스가 집중되지 않게 서비스를 알맞게 분산해 동시에 처리하게 해줍니다. 둘째, 하나의 서버에서 처리한 결과를 다른 서버와도 동기화해서 데이터의 안정성을 보장합니다. 셋째, 운영(active) 서버에 문제가 발생해서 서비스를 제공할 수 없을 경우, 다른 대기 중인 서버를 운영 서버로 바꿔서 서비스가 중지 없이 제공되게 합니다. 넷째, 분산 환경을 구성하는 서버의 환경설정을 통합적으로 관리합니다.

리소스 관리

- YARN(<http://hadoop.apache.org>) 안(YARN)은 데이터 처리 작업을 실행하기 위한 클러스터 자원(CPU, 메모리, 디스크등)과 스케줄링을 위한 프레임워크입니다. 기존 하둡의 데이터 처리 프레임워크인 맵리듀스의 단점을 극복하기 위해서 시작된 프로젝트이며, 하둡2.0부터 이용이 가능합니다. 맵리듀스, 하이브, 임팔라, 타조, 스파크 등 다양한 애플리케이션들은 안에서 리소스를 할당받아서, 작업을 실행하게 됩니다. 안에 대한 자세한 설명은 <http://hadoop.apache.org/docs/stable/hadoop-yarn/hadoop-yarn-site/index.html>을 참고하시기 바랍니다.

- Mesos(<http://mesos.apache.org>) 메소스(Mesos)는 클라우드 인프라스트럭처 및 컴퓨팅 엔진의 다양한 자원(CPU, 메모리, 디스크)을 통합적으로 관리할 수 있도록 만든 자원 관리 프로젝트입니다. 메소스는 2009년 버클리 대학에서 Nexus 라는 이름으로 시작된 프로젝트이며, 2011년 메소스라는 이름으로 변

경했으며, 현재는 아파치 탑레벨 프로젝트로 진행중이며, 페이스북, 에어비엔비, 트위터, 이베이 등 다양한 글로벌 기업들이 메소스로 클러스터 자원을 관리하고 있습니다. 메소스는 클러스터링 환경에서 동적으로 자원을 할당하고 격리해주는 매커니즘을 제공하며, 이를 통해 분산 환경에서 작업 실행을 최적화시킬 수 있습니다. 1만대 이상의 노드에도 대응이 가능하며, 웹 기반의 UI, 자바, C++, 파이썬 API를 제공합니다. 하둡, 스파크(Spark), 스톰(Storm), 엘라스틱 서치(Elastic Search), 카산드라(Cassandra), 젠킨스(Jenkins) 등 다양한 애플리케이션을 메소스에서 실행할 수 있습니다.

데이터 저장

- HBase(<http://hbase.apache.org>) H베이스(HBase)는 HDFS 기반의 컬럼 기반 데이터베이스입니다. 구글의 빅테이블(BigTable) 논문을 기반으로 개발되었습니다. 실시간 랜덤 조회 및 업데이트가 가능하며, 각 프로세스는 개인의 데이터를 비동기적으로 업데이트할 수 있습니다. 단, 맵리듀스는 일괄 처리 방식으로 수행됩니다. 트위터, 야후!, 어도비 같은 해외 업체에서 사용하고 있으며, 국내에서는 2012년 네이버가 모바일 메신저인 라인에 HBase를 적용한 시스템 아키텍처를 발표했습니다.

- Kudu(<http://getkudu.io>) 쿠두(Kudu)는 컬럼 기반의 스토리지로서, 특정 컬럼에 대한 데이터 읽기를 고속화할 수 있습니다. 물론 기존에도 HDFS에서도 파케이(Parquet), RC, ORC와 같은 파일 포맷을 사용하면 컬럼 기반으로 데이터를 저장할 수 있지만, HDFS 자체가 온라인 데이터 처리에 적합하지 않다는 약점이 있었습니다. 그리고 HDFS 기반으로 온라인 처리가 가능한 H베이스의 경우, 데이터 분석 처리가 느리다는 단점이 있었습니다. 쿠두는 이러한 문제점들을 보완하여 개발한 컬럼 기반 스토리지이며, 데이터의 발생부터 분석까지의 시간을 단축시킬 수 있습니다. 클라우드에서 시작된 프로젝트이며, 2015년말 아파치 재단의 인큐베이션 프로젝트로 선정되었습니다.

데이터 수집

- Chukwa(<http://chukwa.apache.org>) 척와(Chuckwa)는 분산 환경에서 생성되는 데이터를 HDFS에 안정적으로 저장하는 플랫폼입니다. 분산된 각 서버에서 에이전트(agent)를 실행하고, 콜렉터(collector)가 에이전트로부터 데이터를 받아 HDFS에 저장합니다. 콜렉터는 100개의 에이전트당 하나씩 구동되며, 데이터 중복 제거 등의 작업은 맵리듀스로 처리합니다. 야후!에서 개발했으며, 아파치 오픈소스 프로젝트로 공개돼 있습니다.

- Flume(<http://flume.apache.org>) 플럼(Flume)은 척와처럼 분산된 서버에 에이전트가 설치되고, 에이전트로부터 데이터를 전달받는 콜렉터로 구성됩니다. 차이점은 전체 데이터의 흐름을 관리하는 마스터 서버가 있어서 데이터를 어디서 수집하고, 어떤 방식으로 전송하고, 어디에 저장할지를 동적으로 변경할 수 있습니다. 클라우드에서 개발했으며, 아파치 오픈소스 프로젝트로 공개돼 있습니다.

- Scribe(<https://github.com/facebook/scribe>) 페이스북에서 개발한 데이터 수집 플랫폼이며, Chukwa와는 다르게 데이터를 중앙 집중 서버로 전송하는 방식입니다. 최종 데이터는 HDFS 외에 다양한 저장소를 활용할 수 있으며, 설치와 구성이 쉽게 다양한 프로그램 언어를 지원합니다. HDFS에 저장하려면 JNI(Java Native Interface)를 이용해야 합니다.

- Sqoop(<http://sqoop.apache.org>) 스쿱(Sqoop)은 대용량 데이터 전송 솔루션이며, 2012년 4월에 아파치의 최상위 프로젝트로 승격되었습니다. Sqoop은 HDFS, RDBMS, DW, NoSQL 등 다양한 저장소에 대용량 데이터를 신속하게 전송하는 방법을 제공합니다. 오라클, MS-SQL, DB2 등과 같은 상용 RDBMS와 MySQL, 포스트그레스큐엘(PostgreSQL)과 같은 오픈소스 RDBMS 등을 지원합니다.

- Hiho(<https://github.com/sonalgoyal/hiho>) 스쿱과 같은 대용량 데이터 전송 솔루션이며, 현재 깃헙(GitHub)에 공개돼 있습니다. 하둡에서 데이터를 가져오기 위한 SQL을 지정할 수 있으며, JDBC 인터페이스를 지원합니다. 현재는 오라클과 MySQL의 데이터 전송만 지원합니다.

- Kafka(<http://kafka.apache.org>) 카프카(Kafka)는 데이터 스트림을 실시간으로 관리하기 위한 분산 메시징 시스템입니다. 2011년 링크드인에서 자사의 대용량 이벤트처리를 위해 개발됐으며, 2012년 아파치 탑레벨 프로젝트가 됐습니다. 발행(publish)-구독(subscribe) 모델로 구성되어 있으며, 데이터 손실을 막기 위하여 디스크에 데이터를 저장합니다. 파티셔닝을 지원하기 때문에 다수의 카프카 서버에서 메시지를 분산 처리할 수 있으며, 시스템 안정성을 위하여 로드밸런싱과 내고장성(Fault Tolerant)를 보장합니다. 다수의 글로벌 기업들이 카프카를 사용하고 있으며, 그중 링크드인은 하루에 1조1천억건 이상의 메시지를 카프카에서 처리하고 있습니다.

데이터 처리

- Pig(<http://pig.apache.org>) 피그(Pig)는 야후에서 개발됐으나 현재는 아파치 프로젝트에 속한 프로젝트로서, 복잡한 맵리듀스 프로그래밍을 대체할 피그 라틴(Pig Latin)이라는 자체 언어를 제공합니다. 맵리듀스 API를 매우 단순화한 형태이고 SQL과 유사한 형태로 설계됐습니다. SQL과 유사하기만 할 뿐, 기존 SQL 지식을 활용하기가 어려운 편입니다.

- Mahout(<http://mahout.apache.org>) 머하웃(Mahout)은 하둡 기반으로 데이터 마이닝 알고리즘을 구현한 오픈소스 프로젝트입니다. 현재 분류(classification), 클러스터링(clustering), 추천 및 협업 필터링(Recommenders/collaborative filtering), 패턴 마이닝(Pattern Mining), 회귀 분석(Regression), 차원 리덕션(Dimension reduction), 진화 알고리즘(Evolutionary Algorithms) 등 중요 알고리즘을 지원합니다. Mahout을 그대로 사용할 수도 있지만 각 비즈니스 환경에 맞게 최적화해서 사용하는 경우가 많습니다.

- Spark(<http://spark.apache.org>) 스파크(Spark)는 인메모리 기반의 범용 데이터 처리 플랫폼입니다. 배치 처리, 머신러닝, SQL 질의 처리, 스트리밍 데이터 처리, 그래프 라이브러리 처리와 같은 다양한 작업을 수용할 수 있도록 설계되어 있습니다. 2009년 버클리 대학의 AMPLab에서 시작됐으며, 2013년 아파치 재단의 인큐베이션 프로젝트로 채택된 후, 2014년에 탑레벨 프로젝트로 승격됐습니다. 현재 가장 빠르게 성장하고 있는 오픈소스 프로젝트 중의 하나이며, 사용자와 공헌자가 급격하게 증가하고 있습니다.

- Impala(<http://impala.io>) 임팔라(Impala)는 클라우드에서 개발한 하둡 기반의 분산 쿼리 엔진입니다. 맵리듀스를 사용하지 않고, C++로 개발한 인메모리 엔진을 사용해 빠른 성능을 보여줍니다. 임팔라는 데이터 조회를 위한 인터페이스로 HiveQL을 사용하며, 수초 내에 SQL 질의 결과를 확인할 수 있습니다. 2015년말 아파치 재단의 인큐베이션 프로젝트로 채택됐습니다.

- Presto(<https://prestodb.io>) 프레스토(Presto)는 페이스북이 개발한 대화형 질의를 처리하기 위한 분산 쿼리 엔진입니다. 메모리 기반으로 데이터를 처리하며, 다양한 데이터 저장소에 저장된 데이터를 SQL로 처리할 수 있습니다. 특정 질의 경우 하이브 대비 10배 정도 빠른 성능을 보여주며, 현재 오픈소스로 개발이 진행되고 있습니다.

- Hive(<http://hive.apache.org>) 하이브(Hive)는 하둡 기반의 데이터웨어하우징용 솔루션입니다. 페이스북에서 개발했으며, 오픈소스로 공개되며 주목받은 기술입니다. SQL과 매우 유사한 HiveQL이라는 쿼리 언어를 제공합니다. 그래서 자바를 모르는 데이터 분석가들도 쉽게 하둡 데이터를 분석할 수 있게 도와줍니다. HiveQL은 내부적으로 맵리듀스 잡으로 변환되어 실행됩니다.

- Tajo(<http://tajo.apache.org>) 타조(Tajo)는 고려대학교 박사 과정 학생들이 주도해서 개발한 하둡 기반의 데이터 웨어하우스 시스템입니다. 2013년 아파치 재단의 인큐베이션 프로젝트로 선정됐으며, 2014년 4월 최상위 프로젝트로 승격됐습니다. 맵리듀스 엔진이 아닌 자체 분산 처리 엔진을 사용하며, HiveQL을 사용하는 다른 시스템들과는 다르게 표준 SQL을 지원하는 것이 특징입니다. HDFS, AWS S3, H베이스, DBMS 등에 저장된 데이터 표준 SQL로 조회할 수 있고, 이기종 저장소간의 데이터 조인 처리도 가능합니다. 질의 유형에 따라서 하이브나 스파크보다 1.5 ~ 10배 빠른 성능을 보여줍니다.

워크플로우 관리

- Oozie(<http://oozie.apache.org>) 우지(Oozie)는 하둡 작업을 관리하는 워크플로우 및 코디네이터 시스템입니다. 자바 서블릿 컨테이너에서 실행되는 자바 웹 애플리케이션 서버이며, 맵리듀스 작업이나 피그 작업 같은 특화된 액션으로 구성된 워크플로우를 제어합니다.

- Airflow(<http://nerds.airbnb.com/airflow>) 에어플로우(Airflow)는 에어비앤비가 개발한 워크플로우 플랫폼입니다. 데이터 흐름의 시각화, 스케줄링, 모니터링이 가능하며, 하이브, 프레스토, DBMS 엔진과 결합하여 사용할 수 있습니다.

- Azkaban(<https://azkaban.github.io>) 아즈카반(Azkaban)은 링크드인에서 개발한 워크플로우 관리도구입니다. 링크드인은 자사의 복잡한 데이터 파이프라인을 관리하기 위해서 아즈카반을 개발했으며, 이를 오픈소스로 공개했습니다. 아즈카반은 워크플로우 스케줄러, 시각화된 절차, 인증 및 권한 관리, 작업 모니터링 및 알람 등 다양한 기능은 웹UI로 제공합니다.

- Nifi(<https://nifi.apache.org>) 나이파이(Niagarafiles, Nifi)는 데이터 흐름을 모니터링하기 위한 프레임워크입니다. 여러 네트워크를 통과하는 데이터 흐름을 웹UI에서 그래프로 표현하며, 프로토콜과 데이터 형식이 다르더라도 분석이 가능합니다. 또한 데이터를 흘러 보낼때 우선 순위를 제어할 수 있습니다. 나이파이는 원래는 미국 국가안보국(NSA)에서 개발된 기술로, NSA의 기술 이전 프로그램인 TTP를 통해서 처음 외부에 공개된 오픈소스 기술입니다.

데이터 시각화

- Zeppelin (<https://zeppelin.incubator.apache.org>) 제플린(Zeppelin)은 빅데이터 분석가를 위한 웹 기반의 분석 도구이며, 분석결과를 즉시 표, 그래프로 제공하는 시각화까지 지원합니다. 아이파이썬(iPython)의 노트북(Notebook)과 유사한 노트북 기능을 제공하며, 분석가는 이를 통해 손쉽게 데이터를 추출, 정제, 분석, 공유를 할 수 있습니다. 또한 스파크, 하이브, 타조, 플링크(Flink), 엘라스틱 서치, 카산드라, DBMS 등 다양한 분석 플랫폼과 연동이 가능합니다. 2013년 엔에프랩의 내부 프로젝트로 시작됐으며, 2014년 아파치 재단의 인큐베이션 프로젝트로 선정됐습니다.

데이터 직렬화

- Avro(<http://avro.apache.org>) RPC(Remote Procedure Call)와 데이터 직렬화를 지원하는 프레임워크입니다. JSON을 이용해 데이터 형식과 프로토콜을 정의하며, 작고 빠른 바이너리 포맷으로 데이터를 직렬화합니다. 경쟁 솔루션으로는 아파치 쓰리프트(Thrift), 구글 프로토콜 버퍼(Protocol Buffer) 등이 있습니다.

- Thrift(<http://thrift.apache.org>) 쓰리프트(Thrift)는 서로 다른 언어로 개발된 모듈들의 통합을 지원하는 RPC 프레임워크입니다. 예를 들어 서비스 모듈은 자바로 개발하고, 서버 모듈은 C++로 개발되었을때, 쓰리프트로 쉽게 두 모듈의 통신 코드를 생성할 수 있습니다. 쓰리프트는 개발자가 데이터 타입과 서비스 인터페이스를 선언하면, RPC 형태의 클라이언트와 서버 코드를 자동으로 생성합니다. 자바, C++, C#, Perl, PHP, 파이썬, 델파이, Erlang, Go, Node.js 등과 같이 다양한 언어를 지원합니다.

참고

출처: <http://blrunner.com/99> [BLRUNNER.COM]

From:
<https://atl.kr/dokuwiki/> - **AllThatLinux!**

Permanent link:
https://atl.kr/dokuwiki/doku.php/hadoop_eco_system_2.0

Last update: **2017/02/02 11:11**

