오픈소스 기반의 빅데이터 플랫폼 구축

2014. 03

목 차

1. 오픈 소스 소프트웨어(Open Source Software:OSS) 정의 및 특징

- 공개소프트웨어에 대한 정의는 '공개(Open)'에 담긴 의미를 정의하는 것. 공개의 실체가 소프트웨어의 본질인 '소스코드'이고. 공개되었다는 상태는 누구나 사용할 수 있다는 것을 의미.

-비용을 지불하지 않는다는 의미보다는 소스코드에 대한 변경 등의 접근권한이 최초 제작자와 동일하게 누구에게나 주어진다는 의미가 더 강하다.

- 공개소프트웨어는 소프트웨어의 내용을 프로그래밍 언어로 나타낸 '소스코드'를 공개하여 누구나 개량·재배포할 수 있는 소프트웨어. [출처: 공개SW포털]

① 자유 배포(Free Redistribution)

② 소스코드 공개(Source Code Open)

③ 2차적 저작물(Derived Works) (허용)

④ 소스코드 수정 제한(Integrity of The Author's Source Code)

⑤ 개인이나 단체에 대한 차별 금지 (No Discrimination Against Persons or Groups)

⑥ 사용 분야에 대한 제한 금지 (No Discrimination Against Fields of Endeavor)

⑦ 라이선스의 배포 (Distribution of License)

⑧ 라이선스 적용상의 동일성 유지 (License must not be specific to a product)

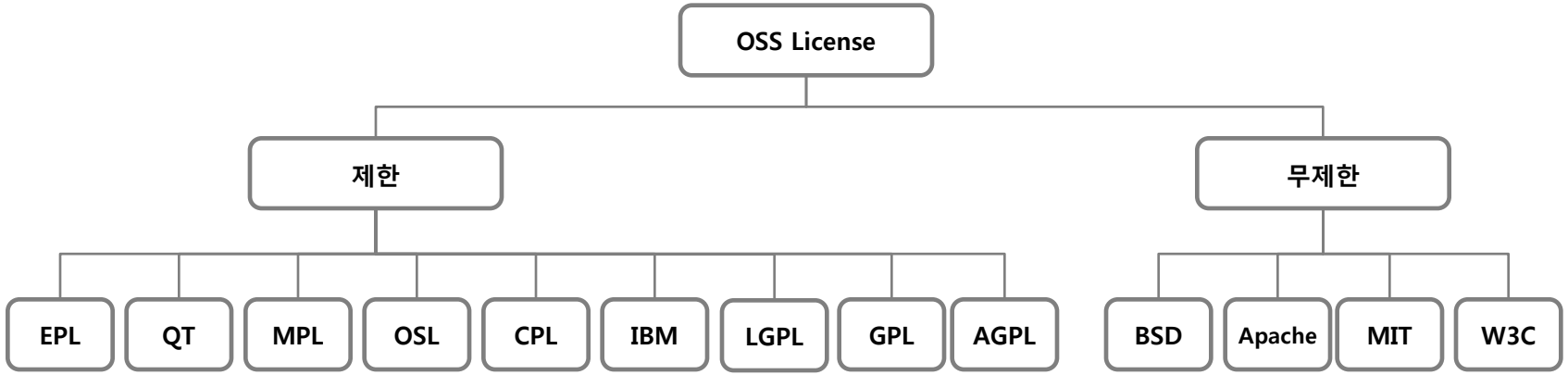
⑨ 다른 라이선스의 포괄적 수용 (License must not contaminate other software)

⑩ 라이선스의 기술적 중립성 (License must be Technology-Neutral)

[출처 : 공개소프트웨어 백서, 2012]

2. 오픈 소스 소프트웨어(Open Source Software:OSS) 라이센스

오픈소스 소프트웨어는 누구나 사용 할 수 있다는 장점이 있으나 특정 라이선스를 갖는 소프트웨어의 경우 2차 가공물을 반드시 공개해야 하는 경우가 있으므로 사용하기 전/후에 꼭 라이선스에 대한 확인이 필요함



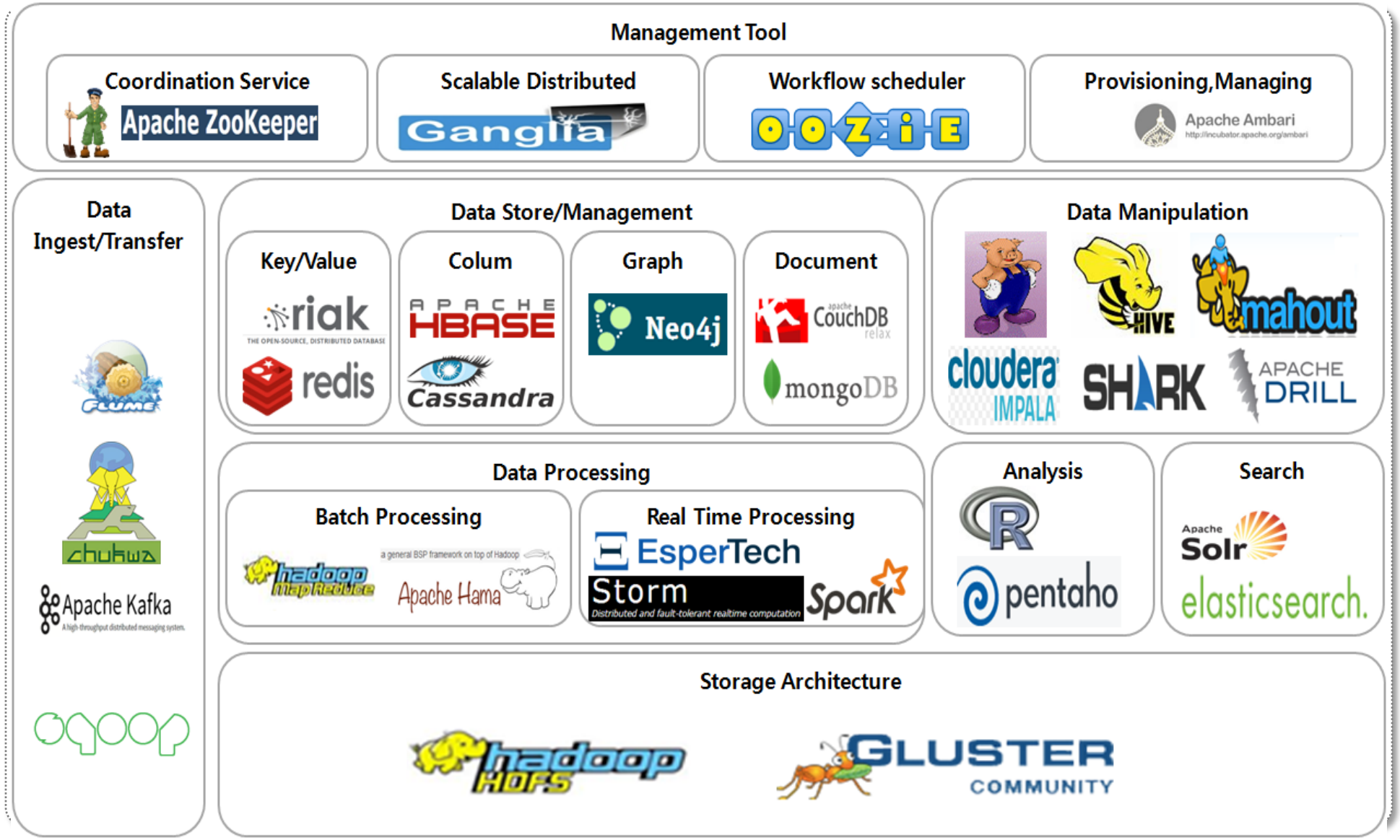
License	배포의 자유	소스코드 공개 및 수정	2차 저작물 재공개 의무	소스코드 수정 제한	사용대상 차별	사용분야 제한	무료이용	독점적 S/W와 결합
GPL/AGPL	허용	가능	**공개**	없음	없음	없음	무료	불가능
LGPL	허용	가능	**공개**	없음	없음	없음	무료	가능
BSD	허용	가능	**비공개 가능**	없음	없음	없음	무료	가능
MPL	허용	가능	**공개**	없음	없음	없음	무료	가능
비공개 소프트웨어	금지	불가능	**금지 혹은 협약하에 가능**	있음	있음	있음	유료	불가능

3. 오픈 소스 소프트웨어(Open Source Software:OSS) 장 단점

공개 소프트웨어와 비공개 소프트웨어는 필요에 따라 적절하게 섞어 쓰는 것이 바람 직 함

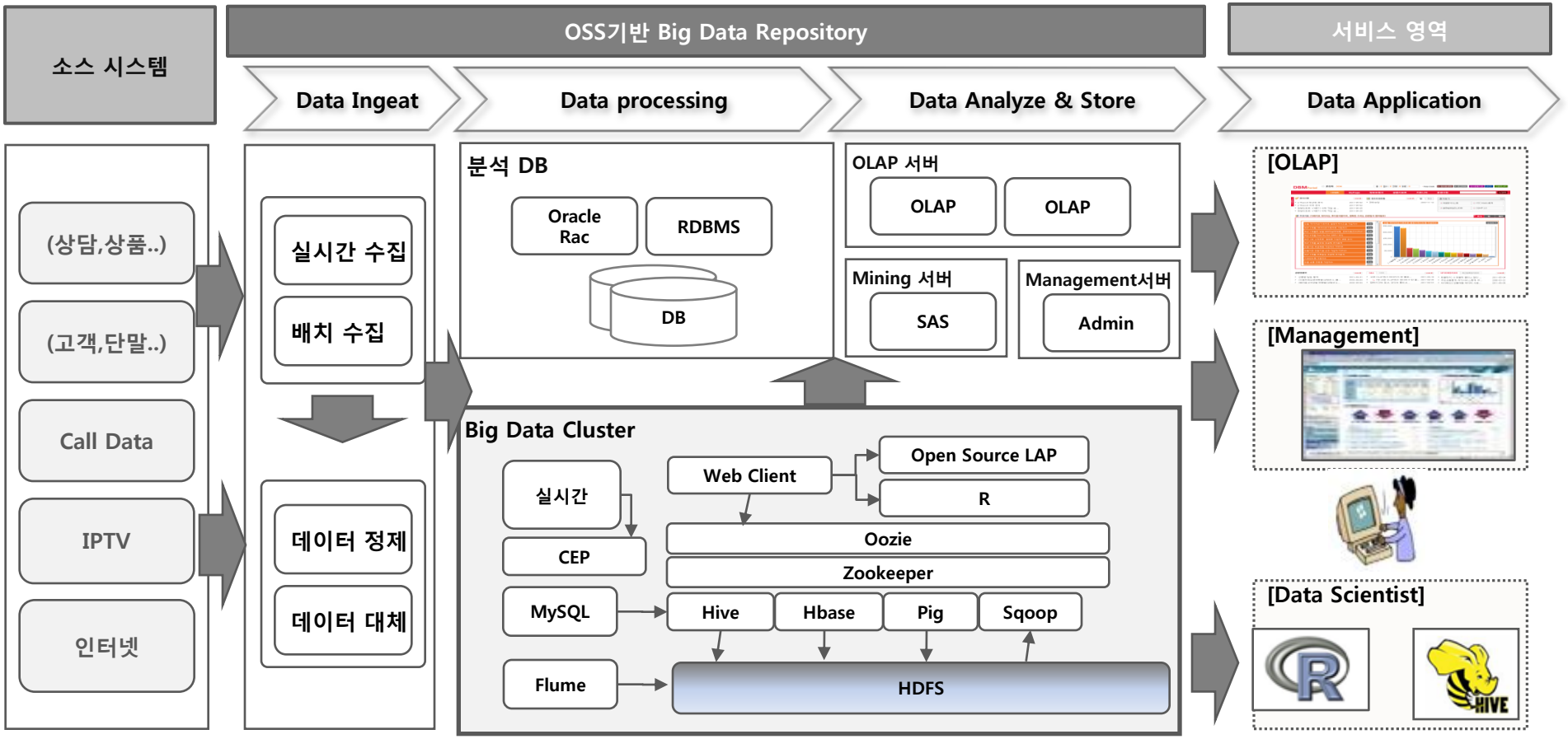
구분	비공개 소프트웨어	공개 소프트웨어
비용	• 도입 비용이 높음(공개 소프트웨어 대비) • 유지보수 비용 및 시스템 개선 비용이 높음 • 소수의 관리자에 의해 유지되고 관리 비용이 높음 • 라이선스 사용료에 대한 비용 발생 • TCO 높음	• 도입 비용이 낮음 • 유지보수 비용 및 시스템 개선 비용이 낮음 • 다수의 자원봉사자에 의해 관리, 관리 비용 낮음 • 라이선스 비용은 무료, 서비스 비용 발생 • TCO 낮음
보안성	• 폐쇄적인 운영으로 인한 공개되지 않은 시스템 취약성 보유 • 프로토콜의 호환이 어려워서 인증체계 취약	• 개발 초기부터 공개되어 많은 취약점이 해결된 안전한 상태 • 공개키 기반의 인증 메커니즘 구현을 위한 통합 패키지 존재 • 다양한 암호화 알고리즘 및 키 관리에 대한 기능 제공
저작권	• 일반 라이선스 • 특정 기업에 Lock-in 가능성	• 다양한 공개 소프트웨어 라이선스 • 특정 기업에 Lock-in 가능성 적음
확장성	• 확장성 보장 • 높은 적용비용과 제한된 시스템 운영환경	• 확장성 보장 안 되는 경우도 많음 • 적용비용이 적게 드나, 시간이 오래 걸리는 경우 있음 • 유지보수 업체가 존재하지 않는 경우 많음
문서화	• 문서 수준 높음	• 문서화 열악함, 다양한 경로에서 사용(적용)관련 문서 확보 가능
하드웨어	• 특정 하드웨어에서만 동작되는 경우 있음	• 대체로 범용 하드웨어에서 동작됨. Scale-out 유리

4. 오픈 소스 소프트웨어(Open Source Software:OSS) 구성

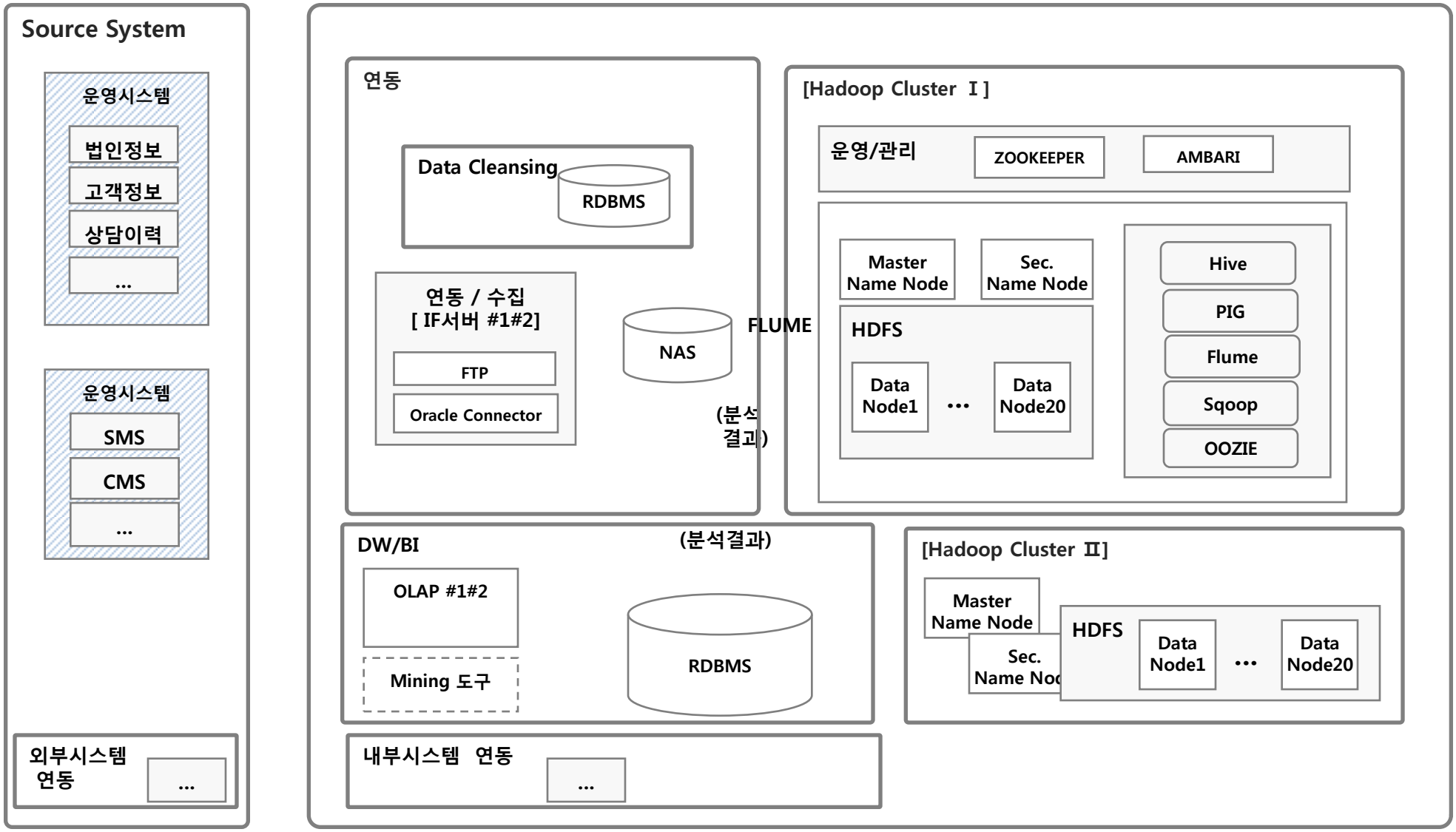


1. 오픈소스 기반의 빅데이터 플랫폼

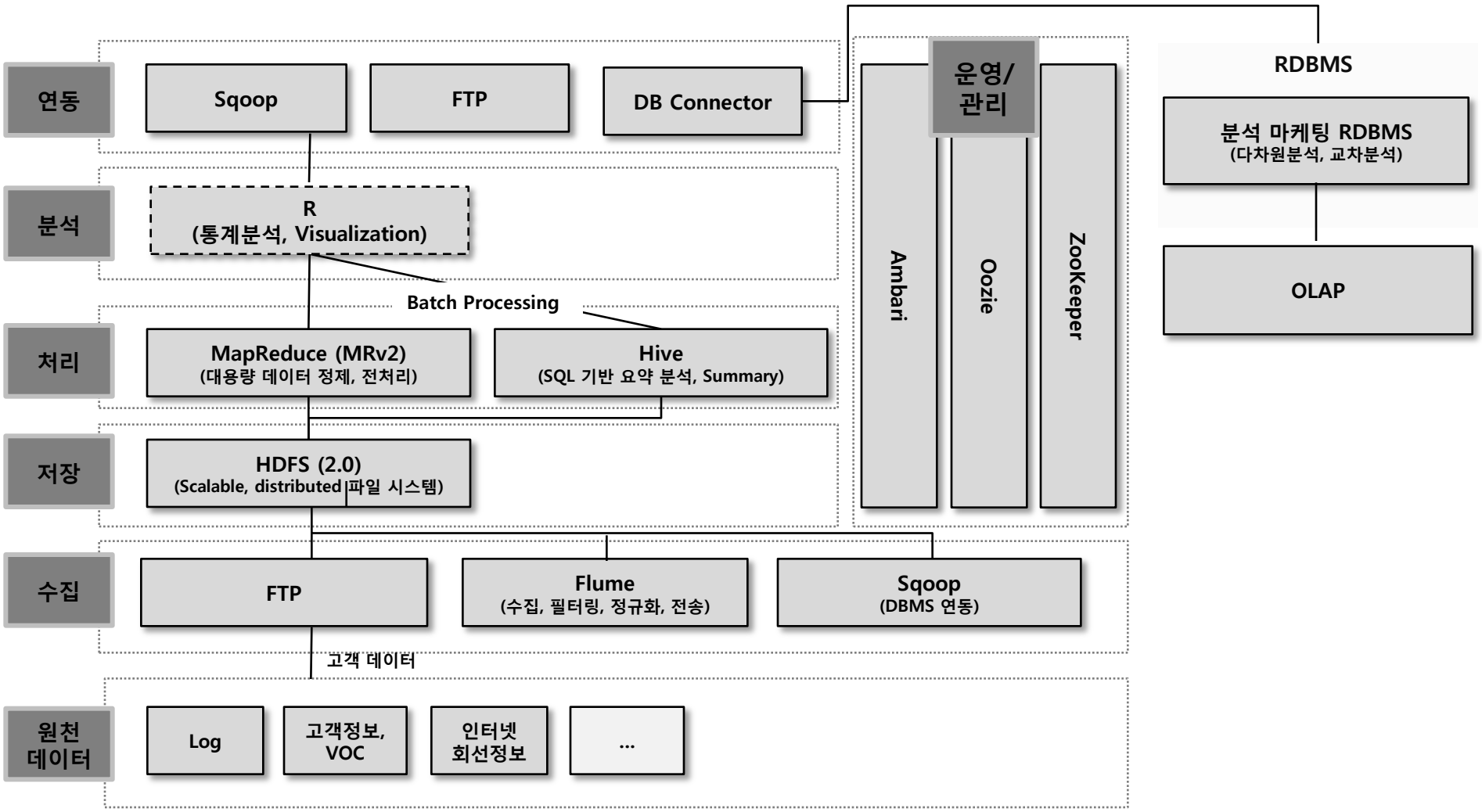
대규모 원본데이터를 대상으로 분석작업이 가능 하도록 Open Sources 및 Hadoop을 이용한 데이터 통합저장소를 구축하고, 대용량 처리 및 상용 솔루션을 통한 분석 환경 구축 함.



2. 오픈소스 기반의 빅데이터 Flow



스토리지, Interface, 관리, 데이터수집 영역에 다수의 프로젝트를 통해 성능과 안정성을 검증한 오픈소스 기술들을 선별해서 사용함



4. 오픈 소스 구성

스토리지, Interface, 관리, 데이터수집 영역에 다수의 프로젝트를 통해 성능과 안정성을 검증한 오픈소스 기술들을 선별해서 사용함

데이터 수집		• 대용량 비정형 데이터를 실시간으로 수집하기 위한 솔루션 • 수집한 데이터를 HDFS 또는 NoSQL에 실시간 저장
데이터 저장		• BDW Data Store • 비정형 데이터를 영구 보관하기 위한 분산 파일 시스템
		• 컬럼 기반의 NoSQL 솔루션 • 비정형 데이터를 정형화 된 데이터로 저장하기 위한 분산 데이터베이스
데이터 처리		• 표준 SQL문을 이용하여 하둡과 NoSQL상의 데이터를 분석할 수 있는 엔진 • DW에서 Ad-hoc 쿼리를 사용할 수 있도록 지원
	PIG 	• 고수준언어로 데이터 분석을 프로그래밍할 수 있는 대용량 데이터셋 분석 플랫폼
		• 원천 데이터 분석 프레임워크 • 분산 파일 시스템에 저장되어 있는 데이터를 이용해 분산 처리를 위한 프레임워크
	Storm 	• 분산 스트리밍 컴퓨팅 기술 • 실시간으로 Stream 타입의 데이터 처리 및 분석
		• 통계 분석/시각화 도구

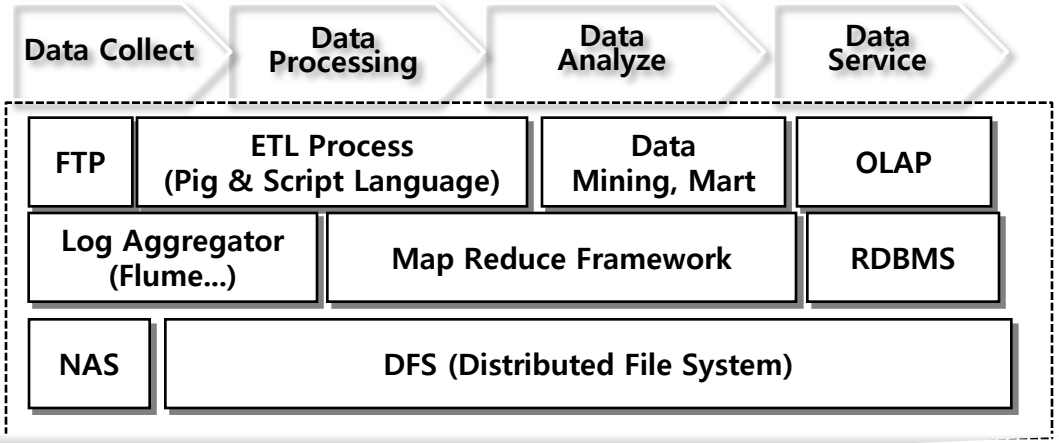
4. 오픈 소스 구성

연동 인터페이스	Sqoop	• SQL to HDFS • RDBMS to HDFS 데이터 마이그레이션 솔루션
운영/관리	Zookeeper	• Service & system coordination • 각 솔루션 및 서비스 멤버십 관리 및 글로벌 큐 서비스
	OOZIE	• Data work flow • 데이터 워크 플로우 엔진 및 도구
	Ambari	• 웹 기반의 빅데이터 플랫폼(하둡, Hbase 등) 관리도구 • 서비스/클러스터 모니터링, 설정 관리 및 프로비저닝 기능 제공

1. 기술 영역 정의

미래 Biz. 및 Big Data 활용을 위한 OSS기반 Big Data Platform을 “Scale-Out 가능한 분산파일 기반 데이터 구조”로 정의하고, Open Source 검증을 위해 기술영역별 검증대상 후보를 선정함

검증 기술 영역 정의 — OSS 기반 Big Data Platform 구축을 위한 Open Source 및 솔루션 기술 영역



검증 대상 후보 선정 — 기준 : 안정성, Ref.多, 旣 검증 여부, 기능/성능 미흡 여부, Apache main project 여부 등 고려

수집

Open Source
- Flume ☑
- Chukwa ☐
- Scribe ☐

상용 Solution
- CDC ☐
- 복제툴 ☐
- EAI ☐

전송

Open Source
- Sqoop ☑
- Hiho ☐

상용 Solution
- ETL Tool ☐

저장

Open Source
- ✓ 분산파일시스템
  - HDFS ☑
  - GlusterFS ☐
- ✓ NoSQL
  - HBase ☐
  - Cassandra ☐
  - MongoDB ☐
- ✓ In-Memory
  - Redis ☐
  - Membase ☐

처리

Open Source
- ✓ 배치
  - MapReduce ☑
  - Pig ☑
- ✓ 실시간
  - Esper ☐
  - Storm ☐
  - S4 ☐

SQL On Hadoop

Open Source
- ✓ 배치 쿼리
  - Hive ☑
- ✓ 실시간 쿼리
  - Impala ☐
  - Tajo ☐

분석

Open Source
- ✓ 통계분석
  - R ☐
- ✓ ETL/OLAP
  - Pentaho ☐
- ✓ 그래프 분석
  - Giraph ☐
- ✓ 기계학습
  - Mahaout ☐

Management

Open Source
- ✓ 모니터링
  - Ambari ☑
- ✓ 워크플로우
  - OOzie ☑
- ✓ 코디네이터
  - Zookeeper ☑

2. 기술 영역 검증(수집/전송)

Source System의 다양한 로그데이터를 수집 및 Data Storage에 전송하는 영역으로, 5개의 후보에 대한 비교/검토 수행(수집 : Flume, Chukwa, Scribe / 전송 : Sqoop, Hiho)

2.1. 수집&전송 영역

영역	Open Source	특징	고려사항	사용여부
수집	Flume	• 다양한 로그데이터 수집(실시간/배치), 모니터링 및 전송프로토콜 지원 • 다양한 운영체제 설치가능(Java 기반) • 단순/유연한 아키텍처 제공(Source, channel, Sink 모델 제공) • 비교대상 Open Source 중 가장 많은 레퍼런스 보유	• Source System에 Flume Agent 설치 필요 (I/F서버에 Flume Agent 설치) • Flume NG(Next Generation, 1.x ver.)활용 ※ Flume OG(Old Generation, 0.9x ver.)에 비해 단순한 아키텍처 제공	
	Chukwa	• 준실시간(분) 로그데이터 수집	• 다양한 플러그인 지원하지 않음 • 지원가능 프로토콜 : HTTP	
	Scribe	• Facebook의 자체 Scaling 작업 목적 설계 • 다양한 로그데이터 수집(실시간/배치), • 하둡과 연동하여 많이 활용되며, 다수의 레퍼런스 보유	• 지원가능 프로토콜 : RPC • 문서화작업이 되어 있지 않음(소스레벨 분석 필요)	
전송	Sqoop	• 다양한 저장소(HDFS, RDBMS, NoSQL, DW 등)에 대용량데이터를 신속하게 전송 • 가장 많은 커넥터를 지원 → 상용 : Oracle, MS-SQL, DB2, Netteza 등 → 오픈소스 : MySQL, PostgresSQL, Hive, Hcatalog 등		
	Hiho	• Sqoop과 같은 대용량 데이터 전송 솔루션(github에서 공개)	• Sqoop보다 적은 수의 커넥터 지원 (Oracle, MySQL, Hive, Salesforce, FTP Server 등)	

2. 기술 영역 검증(저장)

수집된 로그데이터와 로그데이터의 분석/처리결과를 저장하는 영역으로 2개의 후보에 대한 비교/검토 수행(분산파일시스템 : HDFS, GlusterFS)

2.2. 저장 영역(1/2)

영역	Open Source	특징	고려사항	사용여부
분산파일 시스템	HDFS	• 분산파일시스템 중 가장 많이 활용되고, 대부분의 밴더사들이 지원 • Hadoop Ecosystem의 기본으로, Big Data 영역에서 기술표준으로써의 역할을 수행		
	GlusterFS	• 다양한 스토리지를 NFS 또는 Fuse로 마운트해서 사용하는 분산파일 시스템 • HDFS와 다르게 메타데이터를 관리하는 서버(HDFS의 Name node)가 필요없음	• HDFS와 비교하여 MR성능 저하 • 로컬 DISK를 대상으로 DISK I/O 비교시 50% 수준의 성능을 보임	

NoSQL, In-Memory DB 영역의 후보군은 사용을 위한 별도 요건이 없으므로 미 고려(NoSQL : Hbase, Cassandra, MongoDB / In-Memory DB : Redis, Membase)

2.2. 저장 영역(2/2)

영역	Open Source	특징	고려사항	사용여부
NoSQL	Hbase	• 하둡플랫폼을 위한 컬럼기반 데이터베이스로써, HDFS의 약점인 실시간 랜덤 조회 및 데이터 업데이트 기능 지원 • HBase에 있는 테이블들은 하둡에서 동작하는 맵리듀스 잡을 위한 입/출력 제공	• 대량 데이터 관리시 쓰기보다는 읽기 요청이 많은 경우 사용	
	Cassandra	• Hash 알고리즘 이용한 Key-Value구조를 가지며, 확장성과 가용성이 우수함	• 트랜잭션 인덱스 미지원 • 제한적인 검색 • 메모리사용량이 많음(Out of Memory)	
	MongoDB	• Document 기반의 NoSQL로 RDB의 유연한 쿼리 가능하며, 확장성이 뛰어남 • Aggregation 기능이 뛰어나며 자체 MR 기능도 제공	• Join 또는 트랜잭션 처리 불가능 • 데이터 갱신/입력시 DISK에 바로 저장하지 않음(데이터 유실 발생 가능)	
In-Momory DB	Redis	• Key-Value 기반으로 마스터/슬래이브 복제 기능 제공 • hases 및 트랜잭션 등 다양한 기능 제공 • 레퍼런스 다수 보유	• 쓰기보다 읽기 성능이 좋음	
	Membase	• 분산 Key- Value 기반이며, 데이터 양에 상관없이 읽기/쓰기 모두 같은 속도 제공	• 데이터 Update 및 트랜잭션 등 Redis에 비해 활용 기능 적음 • Redis에 비해 레퍼런스 적음	

수집된 로그데이터를 처리하는 영역으로 배치처리를 담당하는 2개의 후보(MapReduce, Pig)을 모두 활용하고, 그 외 후보은 사용을 위한 별도 요건이 없으므로 미 고려 (Esper, Storm, S4)

2.3. 처리영역

영역	Open Source	특징	고려사항	사용여부
처리	MapReduce(배치)	• Map과 Reduce라는 두 함수를 활용하여 대용량 데이터를 분산/병렬처리	• (MRv1) Batch성 데이터처리만 제공하며, Namenode 이중화 취약 • (MRv2) 실시간 처리 지원 및 Namenode HA 구성 가능 ※ 3.Hadoop Version[1.0 vs 2.0] (9P) 참고	
	Pig(배치)	• 대용량 데이터셋을 다루기 위한 스크립트 언어로써, MapReduce보다 생산성 좋음 • 데이터 처리 프로세스를 Pig Latin이라는 자체 언어를 사용하여 정의	• MapReduce보다 성능이 느림 • MapReduce와 상호 보완하여 활용	
	Esper(실시간)	• CEP(Complex Event Process)기반 • 단순한 시스템 구성	• Scale up 아키텍처	
	Storm(실시간)	• 분산 스트리밍 기반 오픈소스 • 다양한 개발언어를 제공하고 Falut-Tolerant 성능 우수 • 다수의 레퍼런스 보유	• 복잡한 시스템 구성	
	S4(실시간)	• 분산 스트리밍 기반 오픈소스 • Scale out 방식으로 노드 추가시 선형적 성능 향상(작업이 모든 노드에 균등분배) • 우수한 확장성 : 단순한 API 활용하여 어플리케이션 개발/배포 가능	• 복잡한 시스템 구성 • 성능/안정성에서 Storm이 보다 우수	

2. 기술 영역 검증(SQL-On-Hadoop)

Hadoop에 저장된 데이터를 개발자/사용자에게 친근한 인터페이스인 SQL을 이용하여 처리하는 영역으로 배치처리를 담당하는 후보(Hive)을 활용하고, 그 외 후보은 사용을 위한 별도 요건이 없으므로 미 고려(Impala, Tajo)

2.4. SQL-On-Hadoop

영역	Open Source	특징	고려사항	사용여부
SQL-On-Hadoop	Hive(배치)	• 하둡 기반 DW 솔루션 중 가장 안정적인 오픈소스로 평가 • HiveQL제공(SQL과 매우 유사)으로 높은 생산성을 가짐 • MapReduce의 확장성, 안정성 등의 특징을 모두 가지며, 다양한 도구와 연동 가능	• Batch 실행으로 실시간 실행에 비해 성능이 떨어짐 • MapReduce와 상호 보완하여 활용 ※ Apache main project로써 지속적으로 성능 개선중(現 Hive 0.13 beta)	
	Impala(실시간)	• MapReduce 방식이 아닌 자체 질의 엔진 보유 • 메모리 기반 실시간 쿼리를 수행(다단계로 실행계획 구성시 중간데이터를 HDFS에 저장하지 않으므로 성능이 좋음)	• 다단계로 실행계획 구성시 중단데이터를 메모리에 임시보관하므로 메모리크기를 넘어서는 질의 수행 불가 • 실제 질의 수행후 에러여부 확인가능(안정성을 보장하지 않음)	
	Tajo(Interactive[1)])	• 분산되어 저장된 데이터에 대해 SQL질의를 분산 수행하여 빠른 처리 가능 • 메모리와 디스크를 모두 활용하는 하이브리드 구조로써 Hive에 비해 성능이 좋음	• 최근 출시(2013)되어 검증되지 않음 • 레퍼런스가 많지 않음	

1) 별도의 엔진을 개발하여 MR을 실행하지 않으므로 batch 방식의 Hive보다는 성능이 좋으나, 실시간 처리개념은 아님
(Hive와 Impala의 중간에 위치하며, 홈페이지에서도 Interactive로 표현)

Hadoop & Hadoop Eco System의 설치 및 운영관련 전반적인 관리영역으로 모니터링, 워크플로우 Management, Coordination 등을 위해 3개의 후보(Ambari, Zookeeper, oozie)을 모두 활용예정

2.5. Management

영역	Open Source	특징	고려사항	사용여부
Mgmt.	Ambari (모니터링)	• 가장 대표적인 하둡 모니터링 도구 • 하둡 클러스터 구성, 모니터링, 관리를 위한 GUI 제공		
	Zookeeper (코디네이터)	• 분산환경 관리를 위한 하둡 오픈소스 • 서버 모니터링, Queue 서버, 메타데이터 저장, 분산락 등 다양한 기능 제공		
	oozie (워크플로우)	• 하둡 작업관리를 위한 워크플로우 시스템 • MapReduce 및 Hive/Pig 작업(job)들로 구성된 워크플로우를 제dj	• 실행관리가 UI기반에서 제어되지 않음 (Control-M 등 UI기반 S/W보다 어려움)	

분석 영역의 후보은 사용을 위한 별도 요건이 없으므로 미 고려(R, Pentaho, Giraph, Mahout)

2.5. 분석

영역	Open Source	특징	사용여부
Analysis	R(통계분석)	• 대표적인 통계 분석 오픈소스 • 분산 처리상의 한계로 Hadoop Ecosystem과 연동하여 사용(Rhadoop, Rhive등)	
	Pentaho (ETL/OLAP)	• 대표적인 ETL 관련 오픈소스 • 각종 RDBMS, Nosql, Hadoop 커넥터 제공 • 오픈소스 및 상용버전 모두 보유	
	Giraph (그래프분석)	• 그래프 데이터 처리 분석을 위한 오픈소스 • 주로 SNS 분석에 활용	
	Mahout (기계학습)	• 하둡 기반의 데이터 마이닝 알고리즘을 구현한 오픈 소스 • 분류, 클러스터링, 추천/협업 필터링, 패턴 마이닝, 회귀 분석, 차원 리덕션 등 주요 알고리즘 지원	

Hadoop 배포판 [HDP2.0 vs. CDH4.5]

- **각각의 Open Source 별로 설치가능하나, 개발/운영환경 구축/관리 용이성을 위해 하둡배포판 활용**
- Apache Foundation, Vendor사에서 소스의 변경없이 설정만을 변경하여 packaging한 Hadoop관련 Open Source의 모음으로, 대표적인 배포판은 CDH(Cloudera), HDP(Hortonworks), MAPR(MapR)등이 있음

Hortonworks(HDP2.0)

- 최신 버전 : HDP2.0(Hadoop 2.0 기반), MRv2(YARN) 최신 버전 패키징
- Ambari(Open Source) 활용한 설치/Monitoring
- 하둡배포판 활용은 유/무로 구분이 없고, 기술지원은 별도 계약으로 가능

Component	Package Version
Apache Flume	flume-ng-1.4.0
Apache Hadoop	hadoop-2.2.0
Apache HBase	hbase-0.96.1
Apache HCatalog	hcatalog-0.12.0
Apache Hive	hive-0.12.0
Hue	hue-2.3.0
Apache Oozie	oozie-4.0.0
Apache Pig	pig-0.12.0
Apache Sqoop	sqoop-1.4.4
Apache Mahout	mahout-0.8.0
Apache Zookeeper	zookeeper-3.4.5

Cloudera(CDH4.5)

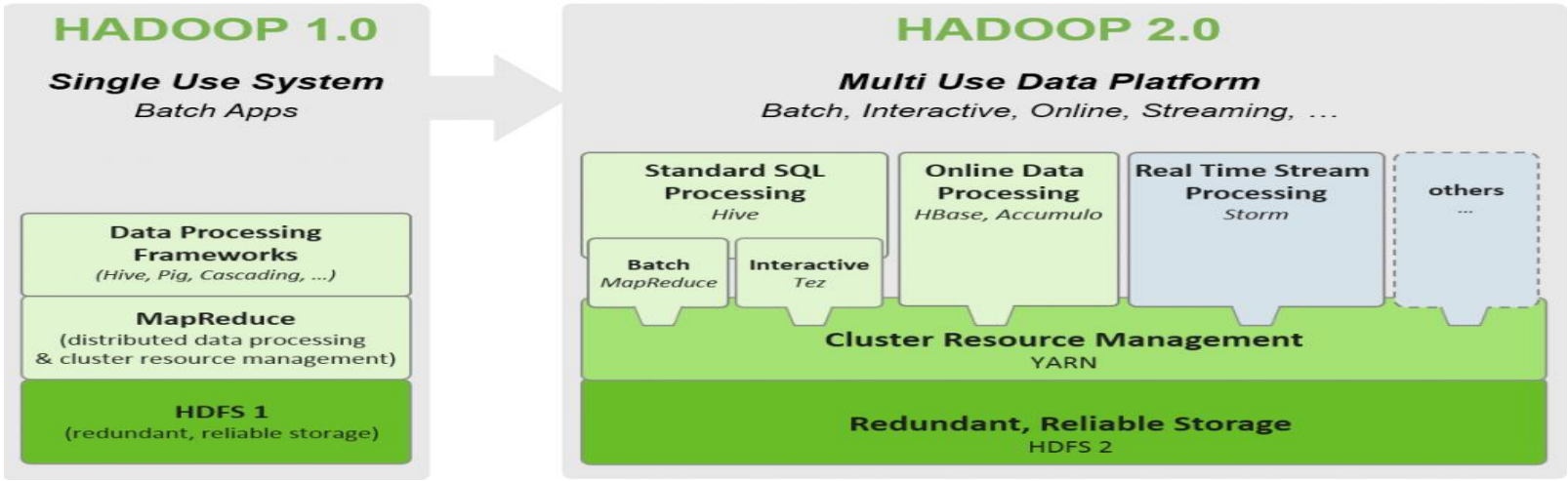
- 최신 버전 : CDH 4.5 stable(Hadoop 2.0 기반/CDH 5.0 beta)
- Cloudera Manager 활용한 설치/Monitoring(무료 : Standard)
- Cloudera Manager Enterprise는 유료 지원

Component	Package Version
Apache Flume	flume-ng-1.4.0+56
Apache Hadoop	hadoop-2.0.0+1518
Apache HBase	hbase-0.94.6+165
Apache HCatalog	hcatalog-0.5.0+14
Apache Hive	hive-0.10.0+214
Hue	hue-2.5.0+182
Apache Oozie	oozie-3.3.2+97
Apache Pig	pig-0.11.0+36
Apache Sqoop	sqoop-1.4.3+81
Apache Mahout	mahout-0.7+22
Apache Zookeeper	zookeeper-3.4.5+24

4. Hadoop Version [1.0 vs 2.0]

Hadoop 2.00이 공식 Release(2013.10) 되었으며 가용성, 확장성 등 고려시 Hadoop 2.0 적용하는 것이 바람직

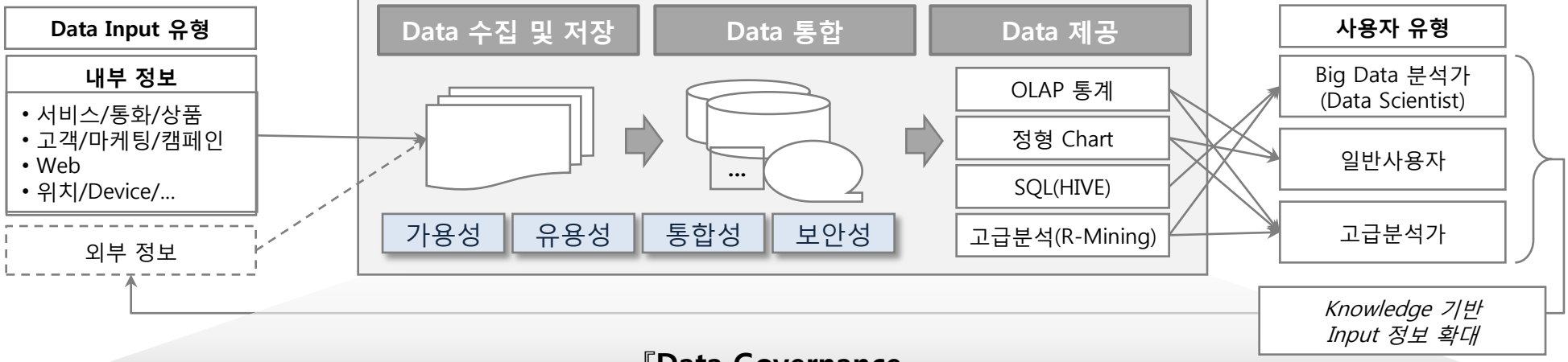
구분	Hadoop 1.0	Hadoop 2.0
특징	• MRv1(MapReduce), HDFS1 • Hadoop 0.20.xx → Hadoop 1.0으로 renumbering	• MRv2, YARN, HDFS2 및 그 외 Application • Hadoop 0.23 → Hadoop 2.0으로 renumbering
	• MapReduce 활용하여 Batch기반 Data Processing 수행	• MapReduce 활용하여 Batch기반 Data Processing 수행 • Interative, Online, Streaming 목적의 Application들을 Hadoop Native App level로 지원가능(다목적 플랫폼) • MRv1 호환.
가용성	• NameNode 이중화 취약	• NameNode HA 지원
확장성		• 대표 vendor사인 Cloudera, Hortonwork, MapR등이 적극적으로 Hadoop2.0을 적용 • 많은 기술들이 Hadoop 2.0을 중심으로 진화 중.



1. Big Data 활용구조

Data를 비즈니스를 위한 핵심 자산으로 최적화하고 내·외부에서 발생하는 다양한 요구에 신속하게 대응하기 위해서 사용하는 Data에 대한 효율적 관리 및 통제, 빠른 적용, 사용자 친화적인 개발 구조가 필요

【OSS기반 Big Data 활용 구조 】



『Data Governance』

1 Data Map 정의

Biz. 관점의 정보 분류

✚

보유 Data 기준 정보 분류

전사 Data Map 정의

"Data 현황 파악 및 Data 활용 목적에 따른 정보 확대 기반"

2 Data 관리 체계 수립

Data 관리 정책	Data 관리 프로세스	조직
• Data 관리 기준 • Data 제공 기준 • 사용자/권한 정의 • 개인정보보호 정책	• Meta 관리 프로세스 • Data 이용 프로세스 • 권환 관리 프로세스	• 조직 구성/유형 • 조직 별 R&R

2. 오픈 소스 소프트웨어 검토 – 기술적/경제적효과

Open Sources Software(OSS) 도입에 따른 SW 구입 비용, 유지보수 비용, 업그레이드 비용, 인력수급 비용 등으로 경제적인 파급 효과를 분석하고 거버넌스, LSB(Linux Standard Base) 인증을 기반으로 개발의 신속성, 핵심 인력 양성, 서비스 품질 제고, 기술 독립성 등 기술적 효과성 검토 함…

※ LSB는 독립 소프트웨어 벤더(ISV)와 개별 어플리케이션 개발자들만을 위하여 만들어진 솔루션이 아니며 리눅스 전체 환경을 위한 표준화 정책 임.

3. 오픈 소스 소프트웨어 검토 – 프로젝트 측면

Open Sources Software(OSS)도입을 위해 플랫폼, 솔루션, 프로젝트 수행, 서비스 운영 등 네 가지 관점에서의 검토가 필요 함.

플랫폼측면

- OSS 도입 이해도
- 시스템 성능 및 안정성

서비스 운영 측면

- 유지보수 및 기술지원
- 지속적인 커뮤니티 참여

솔루션 도입측면

- 솔루션 포팅 적합성
- 관련 기술 표준 준수

프로젝트 수행 측면

- 개발 환경 구성의 적합성
- 개발/설치/테스트 등 프로젝트 기간 준수

빅데이터 데이터웨어하우징

요약 데이터가 아닌 대규모 원본 데이터를 대상으로 분석작업을 수행

Hadoop을 이용하여 다양한 형태/포맷의 데이터 통합 저장소로 활용

Hive기반Ad-Hoc 질의, MapReduce를 이용한 비정형 분석

빅데이터 추천시스템

- 기존추천방식

용량 확장이 용이하지 않고 비용이 많이듬

복잡도나 데이터양이 늘어 날 경우 처리시간이 선형적으로 증가할 수 있음.

- 오픈소스 기반의 빅데이터

HDFS : x86 기반 하드웨어를 사용 하기 때문에 적은 비용으로 용량 확장 가능

MapReduce : 추천을 위한 계산을 여러 컴퓨터에 분산 처리, 빠른 시간에 결과를 얻어낼 수 있음

Mahout : MapReduce기반 추천 알고리즘을 구현하기 위한 라이브러리

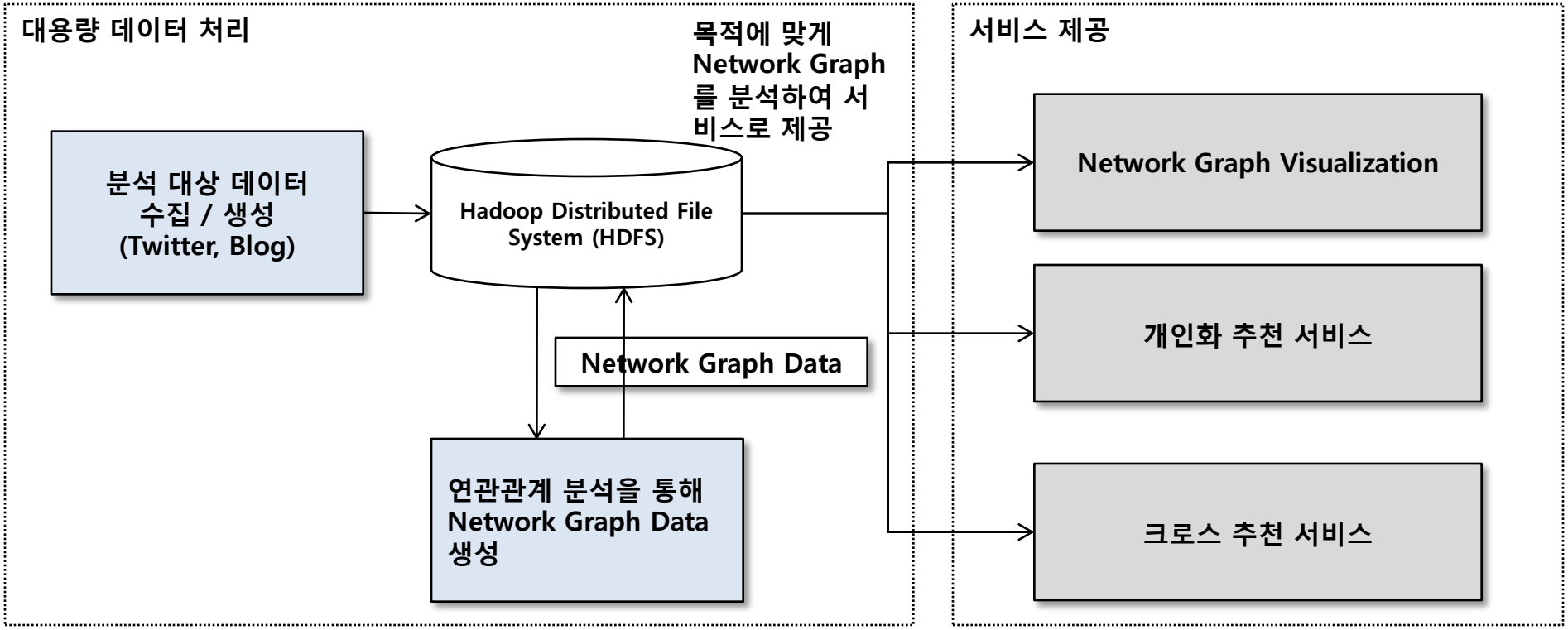
NoSQL : 실시간Query를 통한 데이터 조회 및 분석

SNM (Social Network Mining)

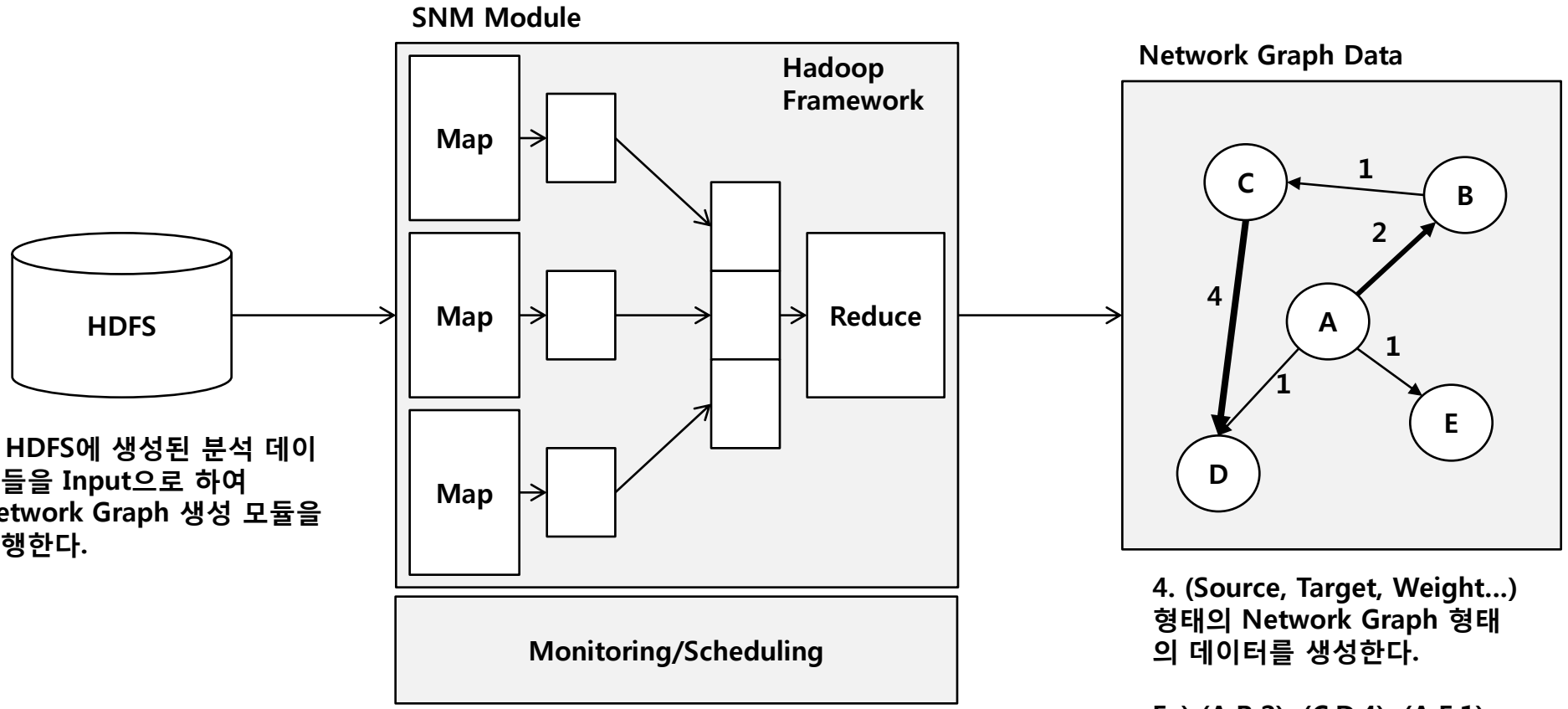
• 인터넷 또는 모바일에서 수집되는 다양한 데이터들을 분석하여 연관 관계를 형성하고 이를 이용하여 마케팅을 위한 새로운 서비스를 창출한다.

• 연관 관계 분석을 위해 기본적으로 *Network Graph*를 활용한다.

• 대상이 되는 데이터를 *Node*라고 하고, Node들 간의 연관 관계를 *Edge*라고 정의한다.
(ex. Twitter의 경우 사용자가 Node, Following/Follower관계가 Edge)



Network Graph Data 생성



1. HDFS에 생성된 분석 데이터들을 Input으로 하여 Network Graph 생성 모듈을 수행한다.

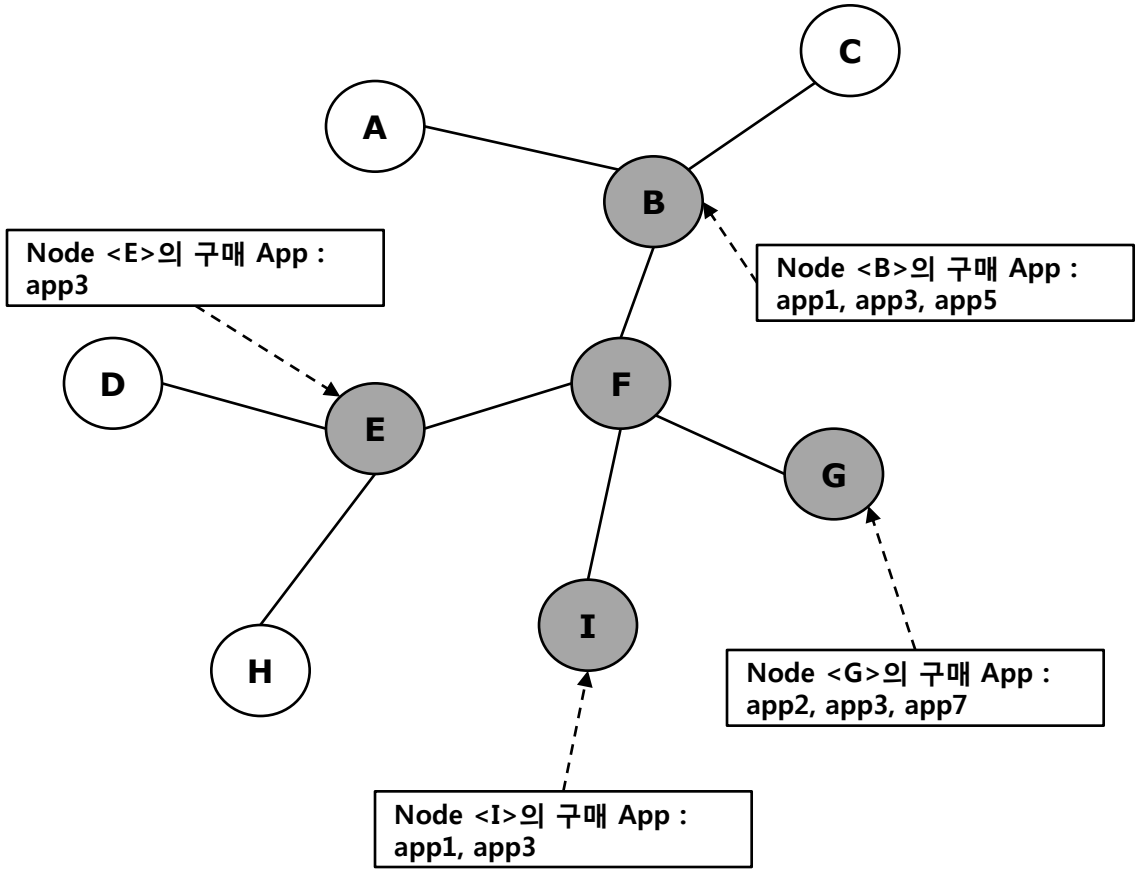
2. Mapper : Edge의 기준이 될 Field를 Key로, Node의 기준이 될 Field를 Value로 하여 Map단계를 수행한다.

3. Reducer : 하나의 Edge에 모인 Node들을 한 쌍씩 출력한다.

4. (Source, Target, Weight…) 형태의 Network Graph 형태의 데이터를 생성한다.

Ex) (A,B,2), (C,D,4), (A,E,1)

개인화 추천 서비스(Recommend by Neighbors)

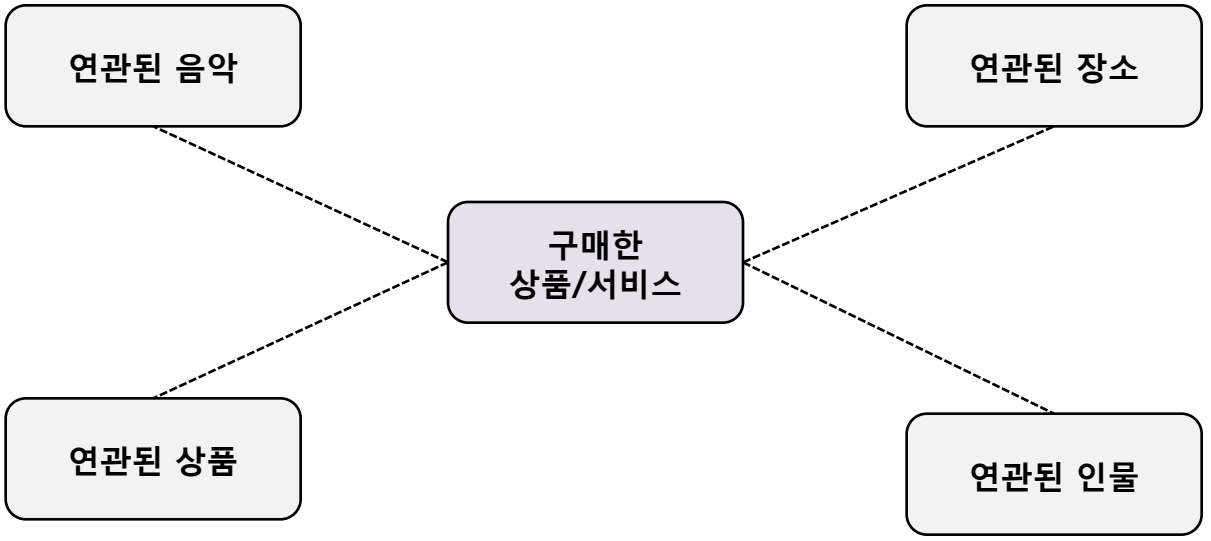


Node <F>가 추천 대상일 경우

- 이웃 Node : B, E, I, G
- 이웃 Node들이 구매한 App List : app1, app2, app3, app5, app7
- App List에서 Node들의 영향력 지수(BPI 등) + App별 이웃 Node가 구매한 횟수에 따라 app 추천 순서가 정해진다.
- app3의 경우 이웃 Node가 많이 구매했으므로 가중치가 높아진다.
- Node <B>가 영향력이 더 높다면 <B>가 구매한 app1, app3, app5의 가중치가 높아진다.
- 추천 하는 App 수가 3개이면 App List에서 점수가 가장 높은 3개의 app만 추천한다.

- **Network Graph Data를 분석하여 유사한 성향을 가진 Node들을 그룹으로 분류한다.**
- **개인이 속해있는 그룹의 특성과 주변 Node간의 영향력을 분석하여 그에 맞는 추천 서비스를 제공한다.**
- **대부분의 데이터 분석은 Hadoop의 Map/Reduce 과정을 통해 수행된다.**

크로스 추천 서비스

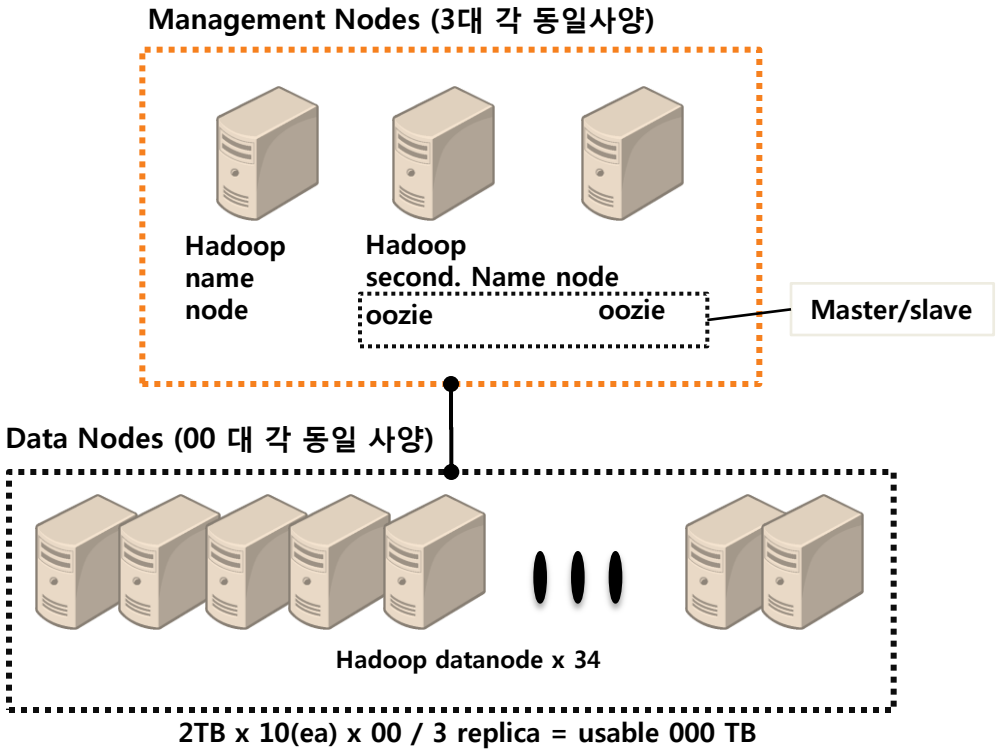


- 수집된 데이터를 이용하여 특정 키워드에 대해 상호 관련성이 있는 상품을 추천하기 위한 서비스
- Visualization을 통해 사용자에게 보여주고, 연관 상품 검색 및 구매 페이지 연결
- ex) 제공하는 영화 서비스를 구매한 사람에게 영화와 관련된 상품(주인공이 입은 옷, 가방, 영화 속의 관광 명소, 음악)들을 추천

[Backup] Big Data Infra

OSS기반 Big Data Node는 Name node를 포함한 3대의 Management Nodes와 데이터 병렬처리를 위한 00대의 Data Node로 구성됨.

【 BigData Node 구성도】



- Data 작업 Control를 위한 Name node (2 대 이중화)와 워크플로우 관리를 위한 oozie 서버를 포함한 3대의 Management Nodes 구성
- 데이터 처리를 위한 00 대의 Data Nodes 구성

Mgmt Node (DELL R520)

. CPU : Intel Xeon E5-2420 1.9Ghz * 2 ea
(15M Cache, 6 Core, 7.2GT/s QPI, Turbo)
. Chipset : Intel C600
. Memory : 128GB (16GB * 8ea, DDR3-1333)
. Disk : SAS 300GB * 6 ea (2.5", 10k rpm)
. NIC : 1Gbps Dual port + 10Gbps Dual port

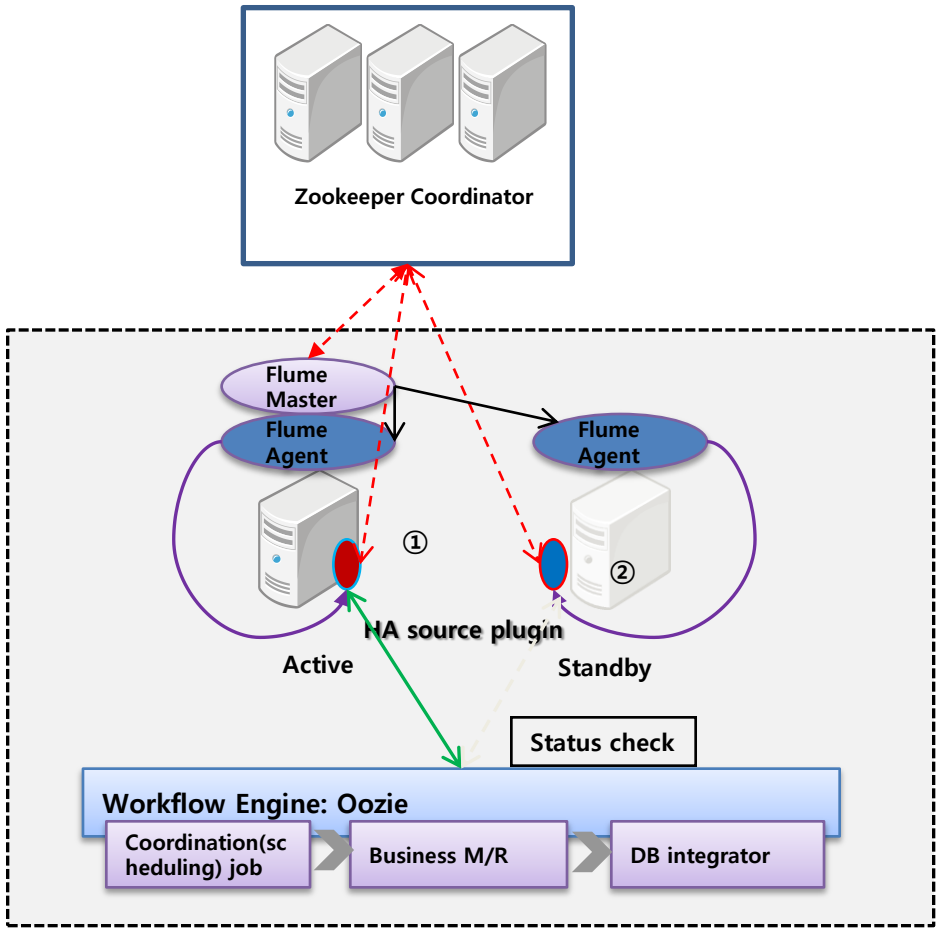
Data Node (DELL R720xd)

. CPU : Intel Xeon E5-2620 2.0Ghz * 2 ea
(15M Cache, 6 Core, 7.2GT/s QPI, Turbo)
. Chipset : Intel C600
. Memory : 64GB (8GB * 8ea, DDR3-1333)
. Disk : SATA 2TB * 12ea (3.5", 7.2k rpm, Max 26 ea)
. NIC : 1Gbps Dual port + 10Gbps Dual port

[Backup] Big Data High Availability(고가용성)

Namenode를 이중화하고, 더불어 Workflow Engine인 Oozie를 Flume기반으로 이중화하고 시나리오를 적용하여 HA를 구성함

【 HA 구성도】



【 HA 구축 내용】

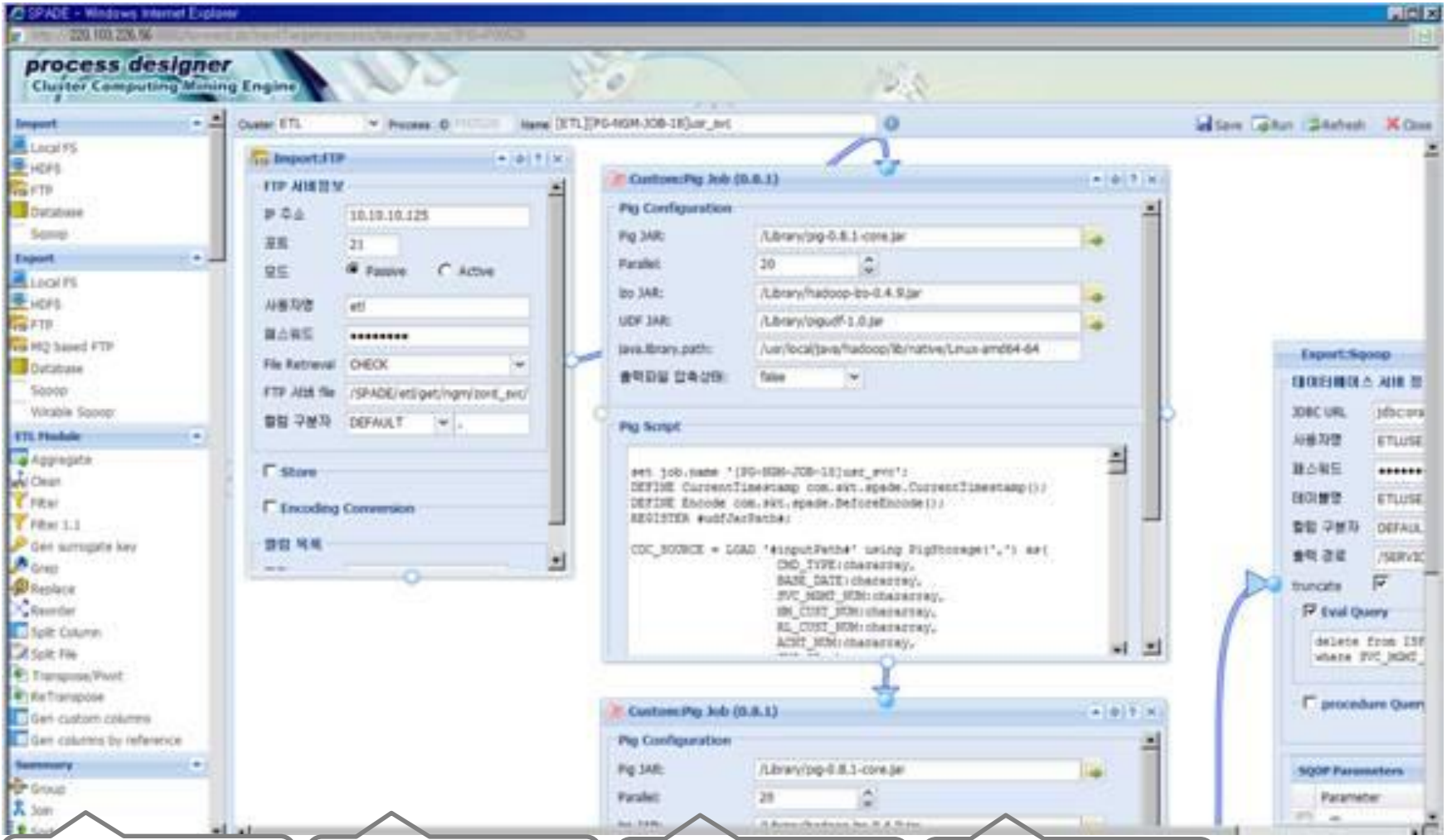
●Flume HA

- Zookeeper Node Master Election
- Zookeeper 이용한 Supervisor Mode
- Google Guide 의 Dependency Injection
  - ✓ Command Pattern 적용
  - ✓ 확장성 용이
  - ✓ 2개의 Class 작성 및 1개의 Invoker로 HA 확장

●Flume HA 시나리오

- ① HA Agent는 3~5초간 oozie 데몬 감시
- ② HA Agent는 대기 및 oozie 데몬 감시
- ① HA Agent 장애가 발생하면, ② HA Agent는 즉시 oozie 데몬 감시하고 Master Election 권한을 가짐.
- 하나의 HA Agent가 감시하는 동안 다른 HA Agent는 Running 모드이지만 작동하지 않음. ➔ Zookeeper ZNode에 lock 발생

[Backup] 관리 도구


process designer
Cluster Computing Mining Engine
HDFS 관리
Job Design
Batch Scheduling
Cluster management

Lessons Learned

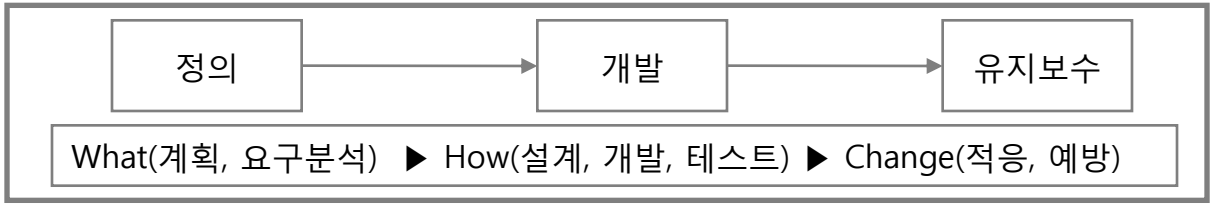
Open sources 기반 Big Data 프로젝트를 진행하면서 ..

- 목표와 달성 수단을 사전에 명확하게 결정하고, 달성에 대한 기준을 설정

garbage in garbage out ⇒ **Big Data in a lot of garbage**

- 기본에 충실

소프트웨어공학 프로세스



소프트웨어 개발 방법론

구성요소	내 용
작업절차 (Process)	- 프로젝트 수행 시 이루어지는 작업단계의 체계 - 단계별 활동, 활동 별 세부작업 열거,활동의 순서 명시
작업방법(Method)	- 각 단계별 수행해야 하는 일에 대한 구체적인 설명 - 절차/작업방법(누가,언제,무엇을 작업하는지를 기술)
산출물	- 각 단계별로 만들어야 하는 산출물의 목록 및 양식
기법	- 각 단계별로 작업 수행 시 소요되는 기술 및 기법 설명
도구	- 각 기법 별 지원도구에 대한 구체적인 사용표준 및 방법

End of Document