데이터분석 전문가 가이드

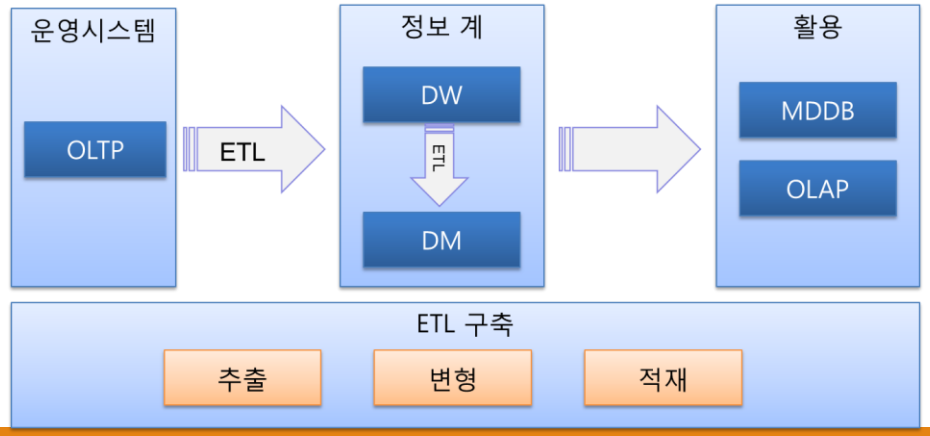
과목 2. 데이터 처리 기술 이해
제 1장 데이터 처리 프로세스

출처 : 데이터분석 전문가 가이드, 한국데이터베이스진흥원

제 1절 ETL(Extraction, Transformation, Load)

1. ETL 개요

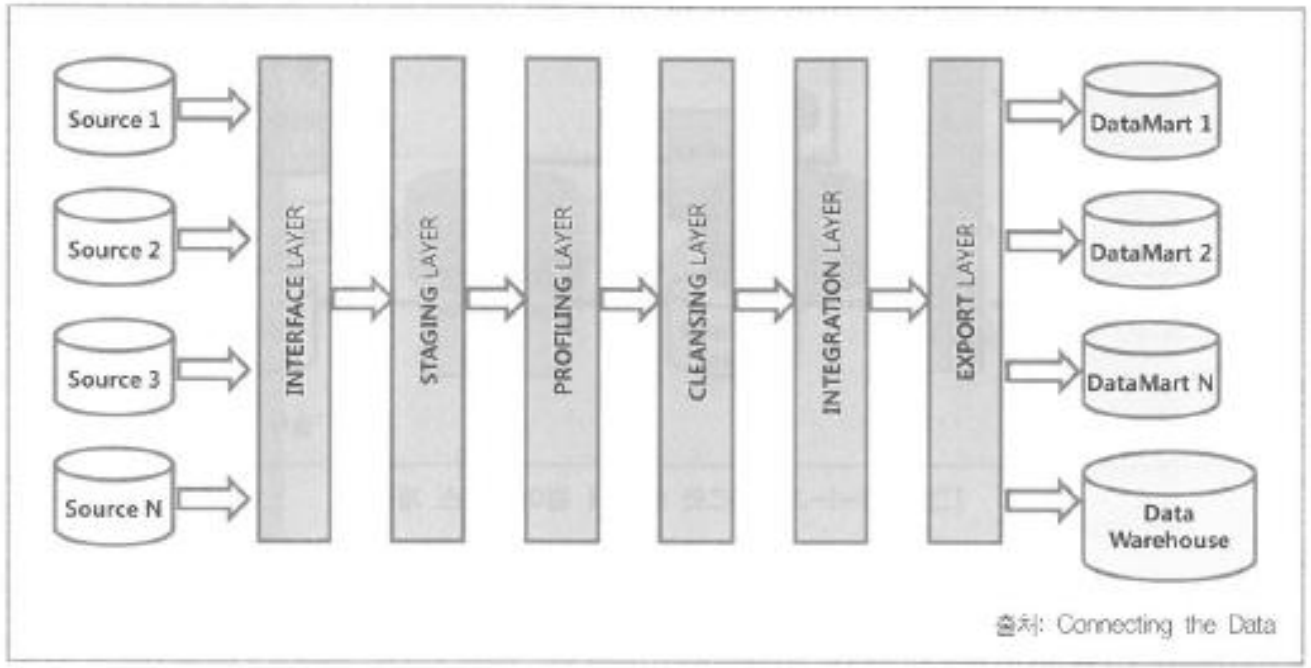
- ETL(Extraction, Transformation, Load)은 데이터 이동과 변환 절차와 관련된 업계표준 용어.
- 데이터 웨어하우스(DW, Data Warehouse), 운영 데이터 스토어(ODS, Operational Data Store), 데이터 마트(DM, Data Mart)에 대한 데이터 적재 작업의 핵심 구성요소로서, 데이터 통합(Data Integration), 데이터이동(Data Migration), 마스터 데이터 관리(MDM, Master Data Management)에 걸쳐 폭넓게 활용.
- ETL의 3가지 기능
  1) Extraction : 하나 또는 그 이상의 데이터 원천들로부터 데이터 획득
  2) Transformation : 데이터 클렌징·형식변환 · 표준화, 통합 또는 다수 애플리케이션에 내장된 비즈니스 룰 적용 등
  3) Loading : 위 변형 단계 처리가 완료된 데이터를 특정 목표 시스템에 적재



제 1절 ETL(Extraction, Transformation, Load)

2. ODS 구성

ODS(Operational Data Store)는 데이터에 추가 작업을 위해 다양한 데이터 원천들로부터 데이터를 추출 ·통합한 데이터베이스.



[그림 II-1-3] Layered ODC Architecture

제 1절 ETL(Extraction, Transformation, Load)

2. ODS 구성

1) 인터페이스 단계 : 데이터를 획득하는 단계로, 데이터 획득을 위한 프로토콜로는 OLEDB, ODBC, FTP 등 있고 , 실시간/근접실시간(Near Real Time)/실시간 데이터 복제 인터페이스 기술이 활용.
2) 데이터 스테이징 단계 : 데이터 원천들로부터 트랜젝션 데이터들을 추출하여 스테이징 테이블에 저장하는 단계로 적재 시간, 데이터 값에 대한 체크섬 등 통계 정보들을 추가함.
3) 데이터 프로파일링 단계 : 데이터 품질을 점검하고 결과 통계/보고서 생성 단계
4) 데이터 클렌징 단계 : 프로파일링 단계에서 식별된 오류 데이터를 수정 단계
5) 데이터 인테그레이션 단계 : 수정 완료된 데이터를 ODS 내의 단일 통합 테이블에 적재 단계
6) 익스포트 단계 : 통합된 데이터를 익스포트 규칙과 보안 규칙을 반영한 익스포트 ETL 기능을 수행해 익스토트 테이블을 생성한 후 데이터 마트 또는 데이터 웨어하우스에 적재

제 1절 ETL(Extraction, Transformation, Load)

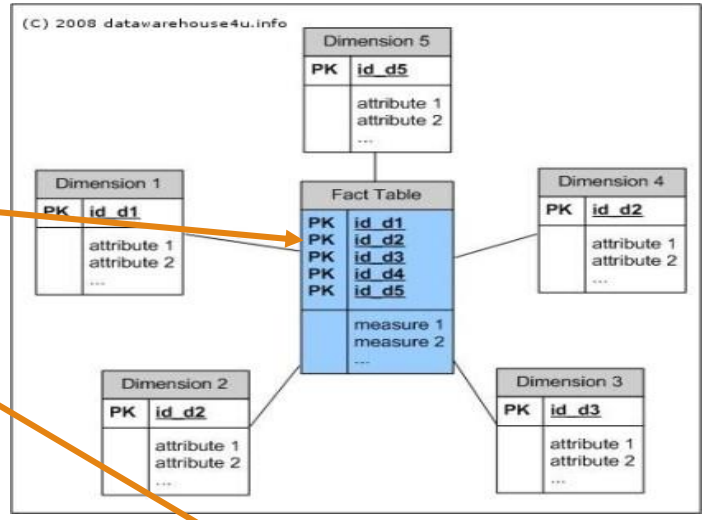
3. 데이터 웨어하우스

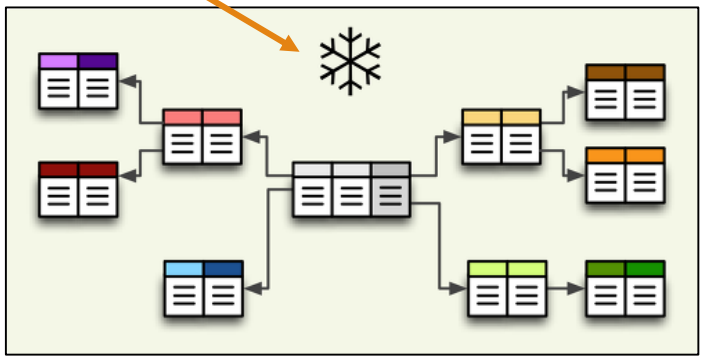
- ODS를 통해 정제 및 통합된 데이터는 데이터 분석과 보고서 생성을 위해 데이터 웨어하우스에 적재
- 데이터 웨어하우스의 특징
  - 주제 중심(Subject Oriented):데이터는 실업무 상황의 특정 이벤트나 업무 환경을 기준으로 구조화
  - 영속성(Non Volatile):최초 저장 이후에는 읽기전용 속성을 가지며 삭제되지 않음
  - 통합성(Integrated):기관 ·조직이 보유한 대부분의 운영 시스템들에 의해 생성된 데이터들의 통합본
  - 시계열성(Time Varient):운영시스템들의 최신 데이터를 보유하고 있지만, 시간순에 의한 이력 데이터를 보유함.

제 1절 ETL(Extraction, Transformation, Load)

3. 데이터 웨어하우스

구 분	Star Schema 모델(반정규화된 모델)	Snowflake Schema 모델(정규화된 모델)
장 점	▪ 모델이 단순하여 사용자 이해가 빠름 ▪ 계층 구조를 정의하기가 쉽다. ▪ 조인 횟수를 줄임으로써 응답 성능 향상 ▪ Meta Data가 단순	▪ 데이터 무결성 유지가 보다 용이 ▪ Star Schema에 비해 상대적으로 작은 저장 공간을 요구 ▪ 애플리케이션의 유연성 증가 ▪ 데이터 중복성 최소화
단 점	▪ Fact 테이블 내의 요약데이터를 추출할 경우 수행속도 저하됨 (계층이 여러 단계인 디멘젼이 반정규화되어 디멘젼이 큰 경우) ▪ 자료의 불일치 위험 ▪ 중복 데이터 포함 ▪ 모델이 유연하지 못함 ▪ 더 많은 저장 공간이 필요	▪ 구조가 복잡해져 Mart 구축의 장점이 희석 ▪ 데이터베이스 내 관리 테이블의 수가 증가 ▪ Dimension 테이블 간의 조인으로 인해 응답성능 저하 ▪ 모델에 대한 사용자의 이해도가 Star Schema에 비해 상대적으로 어려움





제 2절 CDC(Change Data Capture)

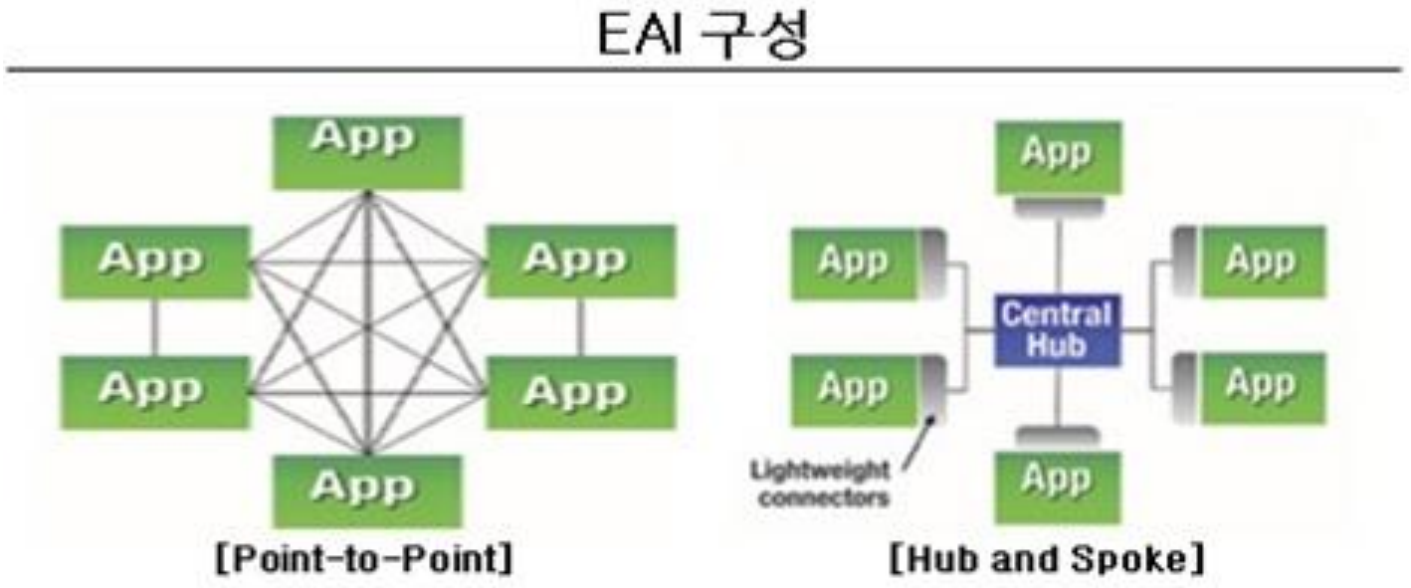
1. CDC 개요

- 데이터베이스 내 데이터에 대한 변경을 식별해 필요한 후속 처리를 자동화하는 기술 또는 설계 기법이자 도구. 실시간 또는 근접 실시간 데이터 통합을 기반으로 하는 데이터웨어하우스 및 기타 데이터 저장소 구축에 활용.
- CDC 구현 기법들
  - Time Stamp on Raws : 테이블 내 마지막 변경 시점을 기록하는 타임스탬프 컬럼을 두고, 마지만 타임스탬프 값보다 더 최근의 타임스탬프 값을 갖는 레코드를 변경된것으로 식별
  - Version Numbers on Raws : 데이블 내 레코드의 버전을 기록하는 컬럼을 둠.
  - Status on Raws : 변경 여부를 True/False 불린 값으로 저장하는 컬럼의 상태 값을 기반
  - Time/Version/Status on Raws : 타임스탬프, 버전 넘버, 상태값의 세 가지 특성을 활용
  - Triggers on Tables : 데이터베이스 트리거를 활용해 사전에 등록된 다수 대상 시스템에 변경 데이터를 배포하는 방식
  - Event Programming : 데이터 변경 식별 기능을 어플케이션에 구현
  - Log Scanner on Database : 데이터베이스의 데이터 변경 여부와 변경 값 시간등을 트랜잭션 로그를 기록/관리하는 기능을 이용하는 방법
- CDC 구현시 푸시(Push) 방식과 풀(Pull) 방식이 있음.

제 3절 EAI(Enterprise Application Integration)

1. EAI 개용

EAI는 기업 정보 시스템들의 데이터를 연계·통합하는 소프트웨어 및 정보 시스템 아키텍쳐 프레임워크로서, 기업 또는 기업 간 복수의 이질적 정보 시스템들의 데이터 연계함으로써 상호 융화 내지 동기화돼 동작하도록 함.



제 3절 EAI(Enterprise Application Integration)

2. EAI 구현 유형

- Mediation(intra-communication) : EAI엔진에 중개자(Broker)로 동작하여, 데이터에 대한 이벤트를 식별해 사전 약속된 시스템들에게 그 내용을 전달(Publish/subscribe Model)
- Federation(inter-communication): EAI엔진이 외부 정보 시스템으로부터 데이터 요청들을 일괄적으로 수령해 필요한 데이터를 전달(Request/reply Model)

3. EAI 기대 효과

- 향후 정보 시스템 개발 및 유지 보수비용 절감 도모
- 기업 업무 정보 시스템들의 지속적 발전 기반 확보
- 협력사 ·파트너 ·고객과의 상호 협력 프로세스 연계 발전 기반 확보
- 웹 서비스 등 인터넷 비즈니스를 위한 기반 토대

제 4절 데이터 연계 및 통합 기법 요약

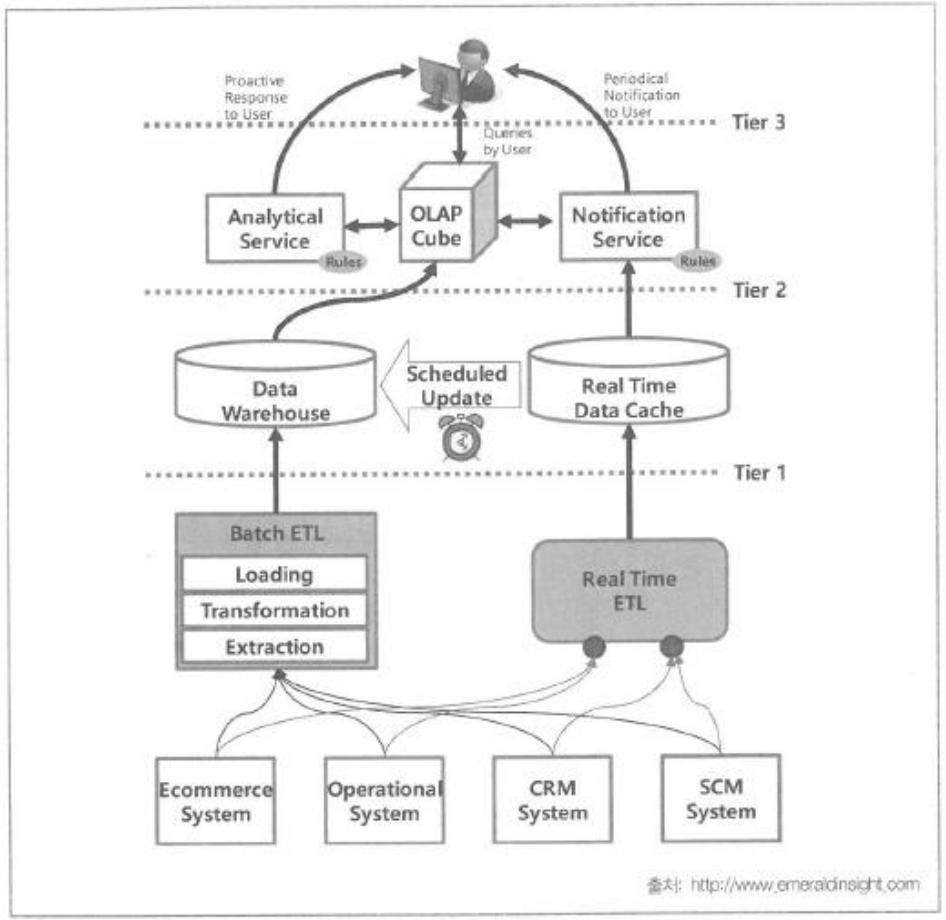
1. 데이터 연계 및 통합 유형(동기화 기준)

[표 II-1-1] 데이터 연계 및 통합 아키텍처 비교

일괄(Batch) 통합	비동기식 실시간 통합	동기식 실시간 통합
• 비실시간 데이터 통합 • 대용량 데이터 대상 • 높은 데이터 조작 복잡성 • 데이터 추출 • 데이터 변형 • 데이터 적재 • CDC(Change data capture) • 감사 증적 • 웹 서비스/SOA • 교차 참조 • 데이터 재 처리 허용 • 점대점 데이터 연계 • 자동화 도구 및 자체 개발 SW 혼용	• 근접 실시간(Near Real Time) 데이터 통합 • 중간 용량 데이터 • 중간 데이터 조작 복잡성 • 데이터 추출·변형·적재 • CDC(Change data capture) • Data pooling and DB Streams • 웹 서비스·SOA • 감사 증적(audit trail) • 교차 참조 • 다수 데이터 원천 및 목표 시스템 • 데이터 재 처리 허용 • 자동화 도구 및 자체 개발 SW 혼용	• 실시간(Real Time) 데이터 통합 • 목표 시스템 데이터 처리 가능 시에만 원천 데이터 획득 • 데이터 추출·변형·적재 • 웹 서비스·SOA • Single transaction integrations • 단일 트랜잭션 단위 데이터 통합 • 데이터 재처리 불가 • 단일 또는 다수 데이터 원천 • 감사 증적

컨테이너 터미널, 공장등의 생산 .운송 장비에서 데이터 수집 IoT

제 4절 데이터 연계 및 통합 기법 요약



[그림 II-1-14] 데이터 연계 및 통합 아키텍처 종합

제 4절 데이터 연계 및 통합 기법 요약

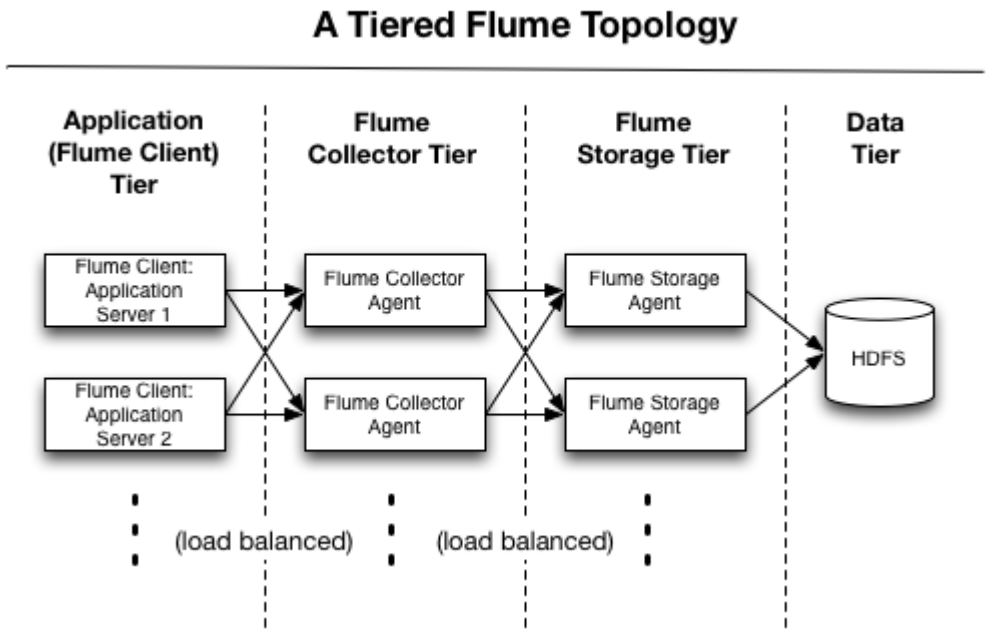
[표 II-1-2] 데이터 처리 기법 비교

구분	전통적 데이터 처리 기법	빅데이터 처리 기법	비고
추출	• 운영 DB(Operational Database)→ ODS • ODS→ 데이터 웨어하우스	• 빅데이터 환경 → 빅데이터 환경	특정 소스에서 타깃 으로 데이터를 옮긴다는 측면은 동일
변환	O	O	
로딩	O	O	
시각화	X	O	시각화를 통해 대용량 데이터에서 통찰력(Insight)을 획득하고자 하는 시도는 빅데이터의 고유한 특성임
분석	• OLAP • 통계(Statistics)와 데이터 마이닝 기술	• 통계(Statistics)와 데이터 마이닝 기술	각종 통계 도구·기법과 데이터 마이닝의 분석 모델 설계·운영·개선 기법의 적용은 유사함
리포팅	비즈니스 인텔리전스	비즈니스 인텔리전스	
인프라스트럭처	• SQL • 전통적 RDBS 인스턴스 (HA 포함)	• NoSQL 등 • 초대형 분산(Redundant) 데이터 스토리지	전통적 데이터 저장 메커니즘 대비 매우 다수의 노드에 중복을 허용하는 방식으로 데이터를 저장하는 것은 빅데이터의 고유한 특성임

제 5절 대용량 비정형 데이터 처리

1. 대용량 로그 데이터 수집

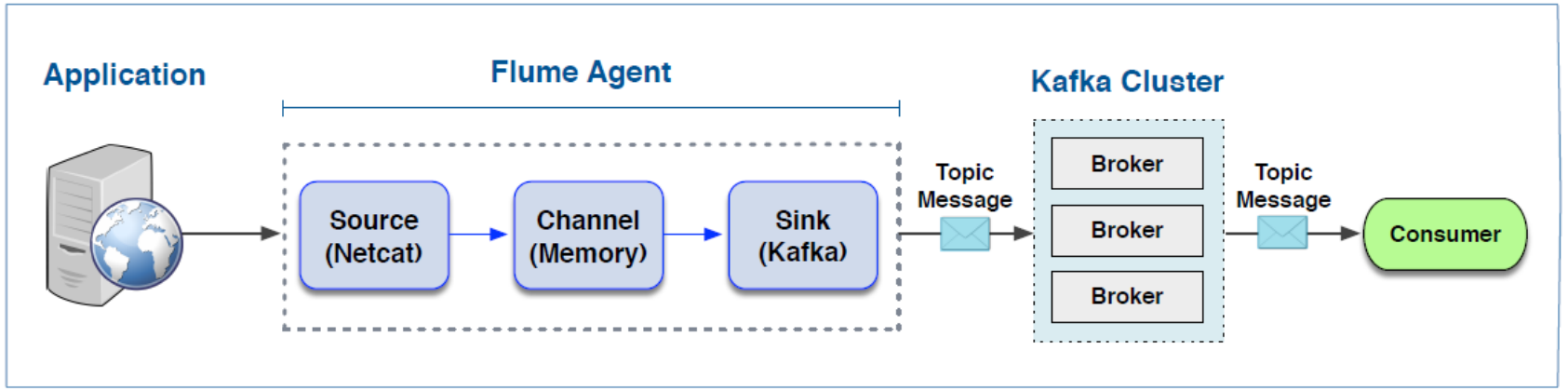
- 초고속 수집 성능과 확정성
- 데이터 전송 보장 메커니즘
- 다양한 수집과 저장 플러그인
- 인터페이스 상속을 통한 애플리케이션 기능 확장



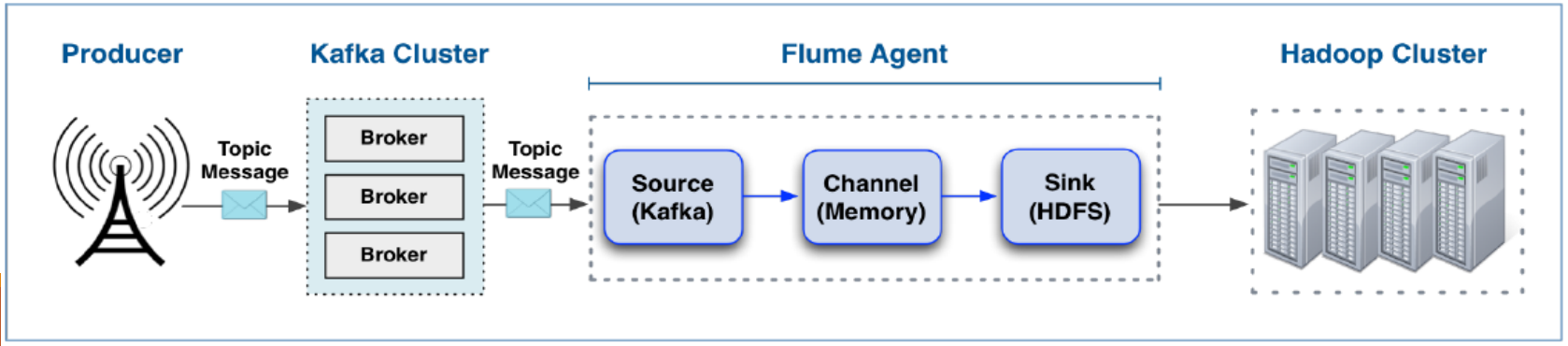
제 5절 대용량 비정형 데이터 처리

1. 대용량 로그 데이터 수집

Using a Flume Kafka Sink as a Producer



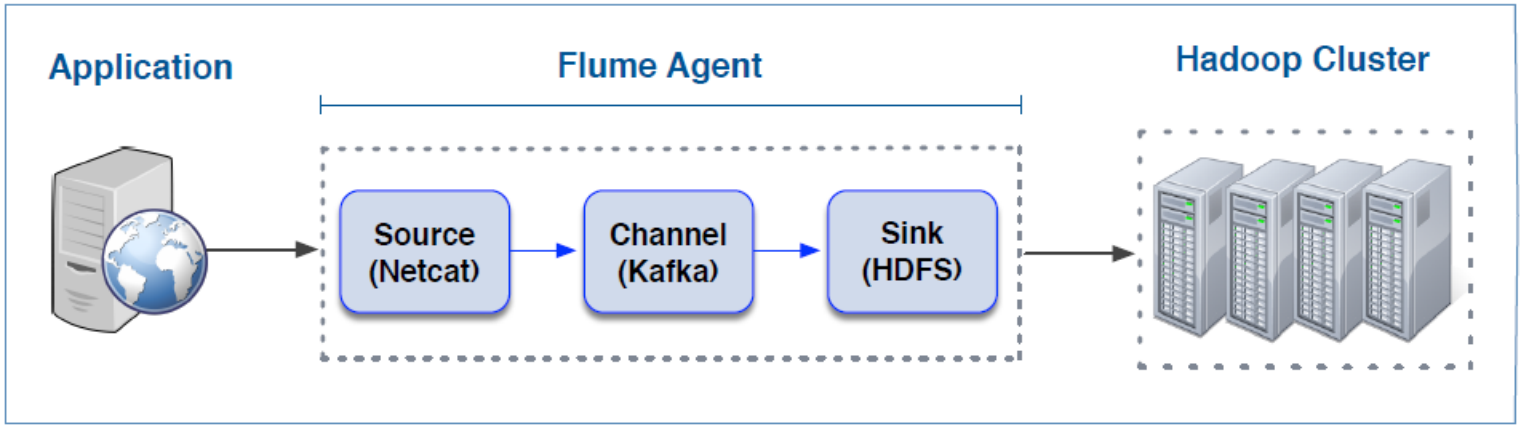
Using a Flume Kafka Source as a Consumer



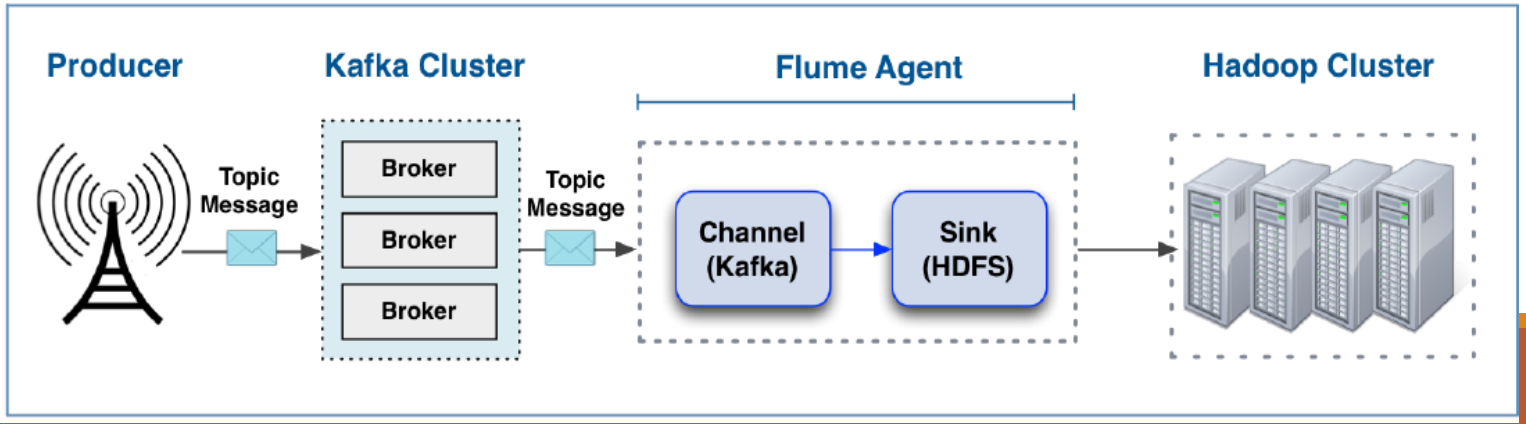
제 5절 대용량 비정형 데이터 처리

1. 대용량 로그 데이터 수집

Using a Flume Kafka Channel



Using a Kafka Channel as a Consumer (Sourceless Channel)

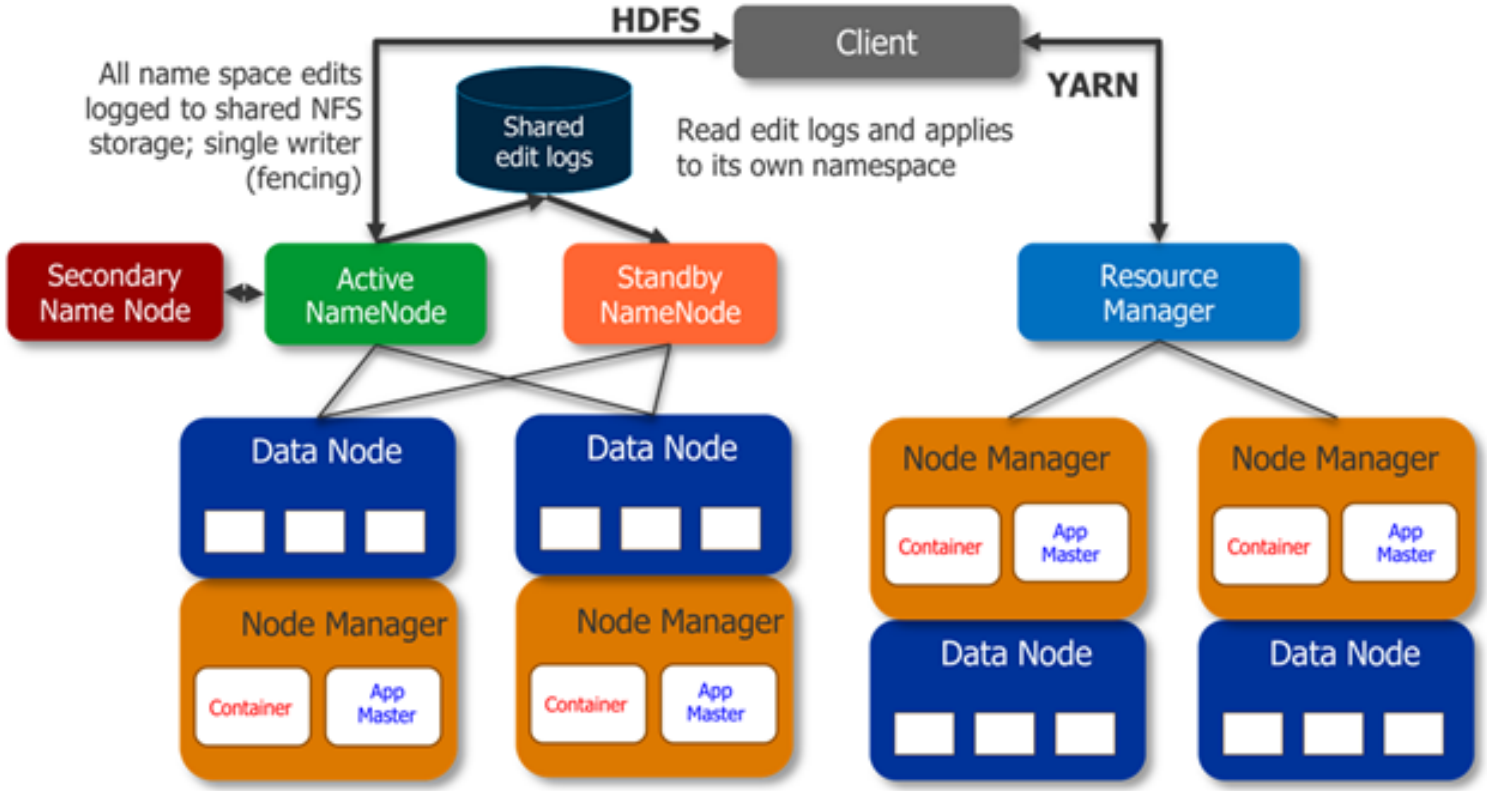


제 5절 대용량 비정형 데이터 처리

2. 대규모 분산 병렬 처리

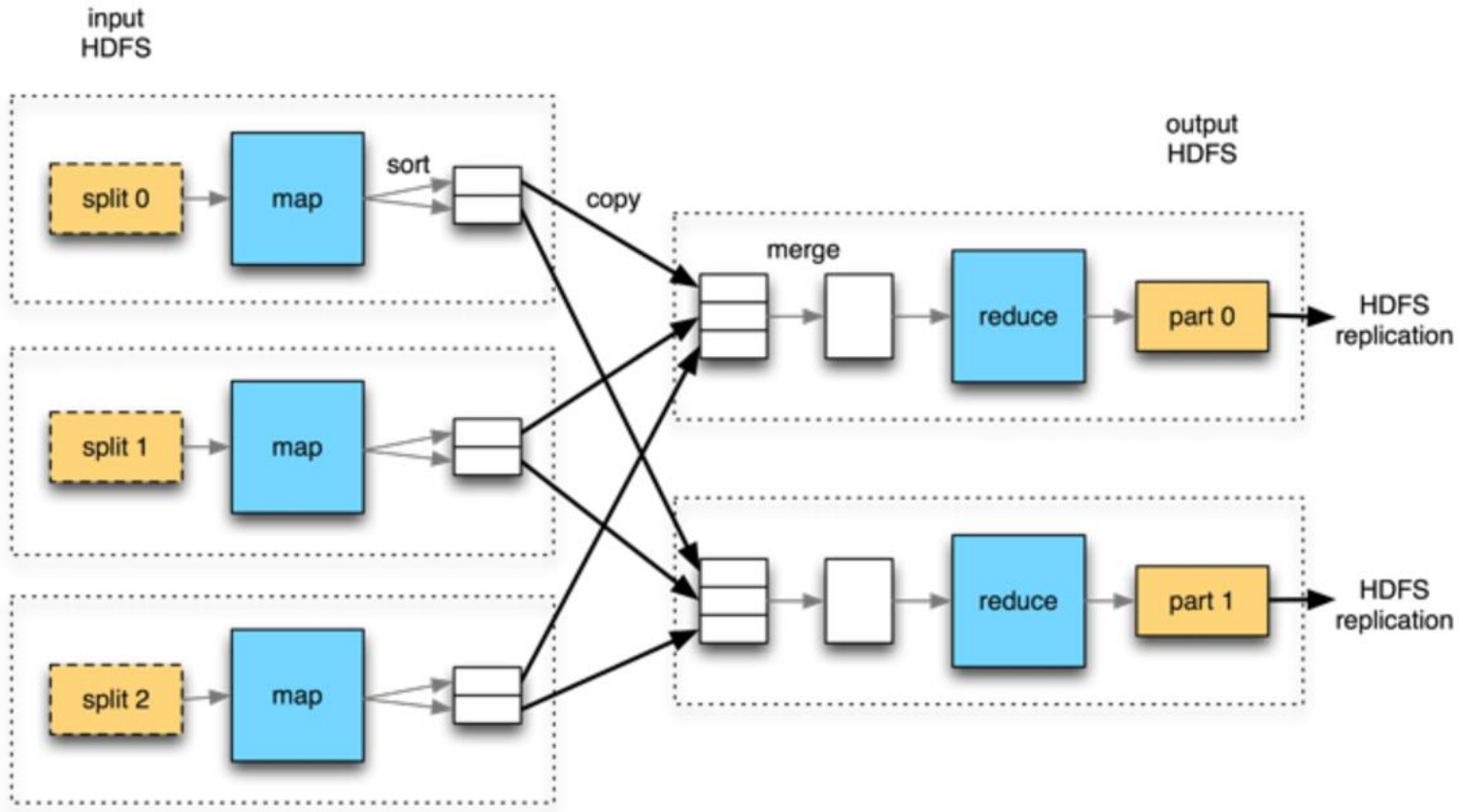
- 선형적인 성능과 용량 확장
- 고장 감내성
- 핵심 비즈니스 로직에 집중
- 풍부한 에코 시스템 형성

하둡 2.x 아키텍쳐



제 5절 대용량 비정형 데이터 처리

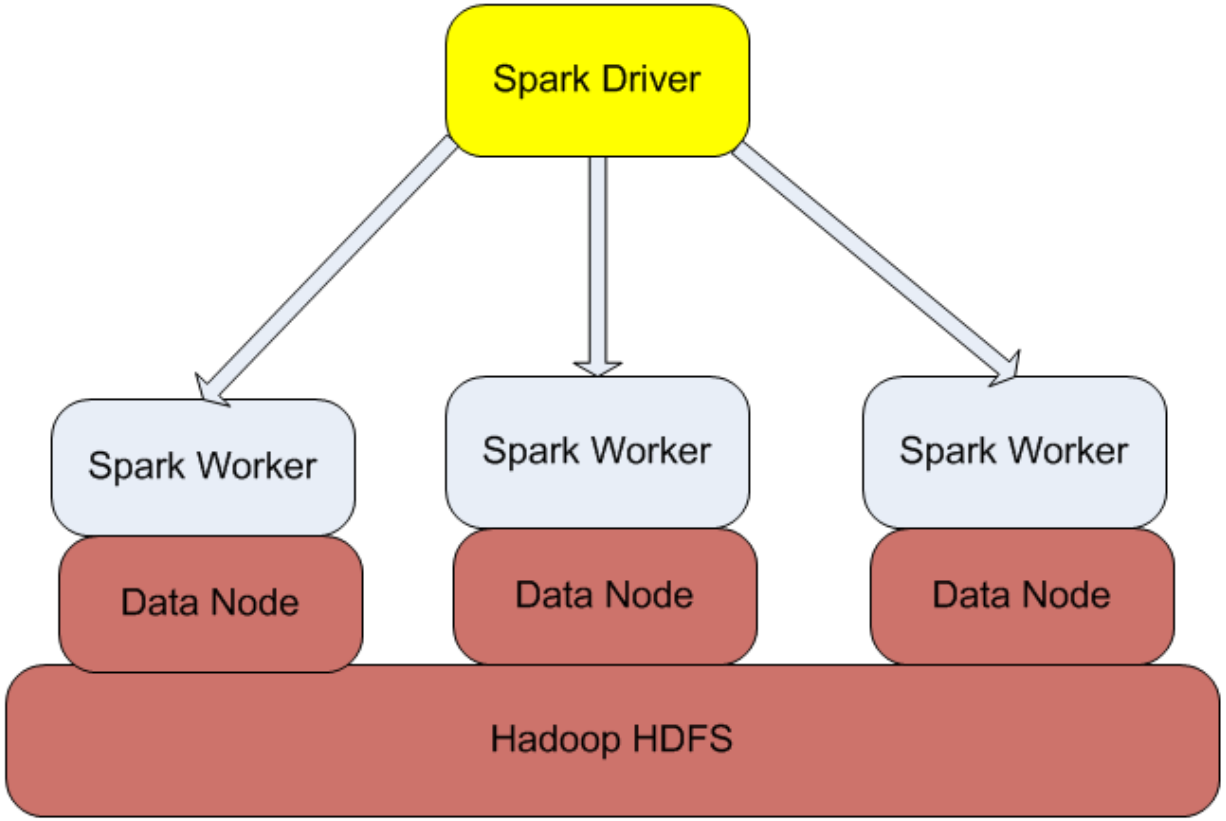
2. 대규모 분산 병렬 처리



제 5절 대용량 비정형 데이터 처리

2. 대규모 분산 병렬 처리

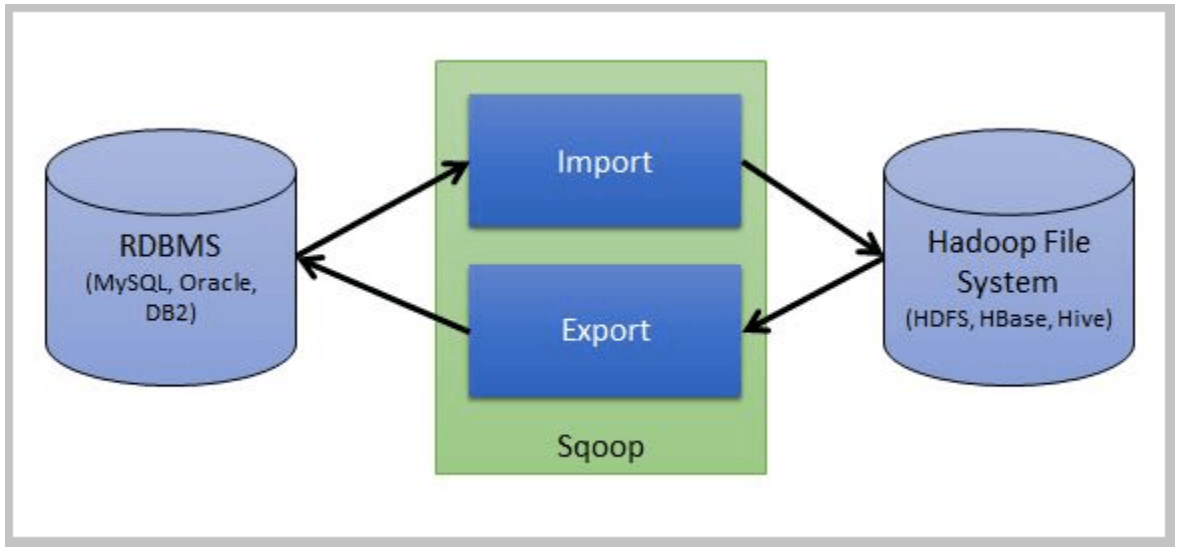
Hadoop + Spark 혼합



제 5절 대용량 비정형 데이터 처리

3. 데이터 연동

```
sqoop import \
  --connect jdbc:mysql://localhost/loudacre \
  --username training --password training \
  --table accounts \
  --target-dir /loudacre/accounts_parquet \
  --as-textfile or
  --as-parquetfile   or
  --as-avrodatafile
```



제 5절 대용량 비정형 데이터 처리

4. 대용량 질의 기술

- 자바와 같은 프로그램 언어를 사용하지 않고, SQL 이라는 질의기술로 이용하여 하둡상의 저장된 데이터를 쉽게 처리하는 분석하는 도구인 하이브(Hive) 가 있음.
- 하이브는 대용량 데이터를 배치처리에 최적화되어 있고, 실시간 조회에는 제약이 많음.
- 최근에는 “SQL on 하둡”이라는 실시간 SQL 질의 분석 기술이 나옴.
- SQL on 하둡
  - 아파치 드릴(Drill), 아파치 스팅커(Stinger), 샤크(Shark), 아파치(Tajo), 임팔라(Impala),…
- SQL on 하둡 관련 기술들은 인메모리 방식으로 성능은 좋으나, 수TB 이상 데이터처리시 문제를 발생하고 있음.