Hive

아꿈사
Cecil

Contents

- 하이브란?
- 하이브 설치하기
- 하이브의 컨포넌트
- 전통적인 데이터 베이스와의 비교
- HiveQL
- 테이블
- 데이터 쿼리하기
- References

하이브란?

하둡에 저장된 데이터를 쉽게 처리할 수 있는
데이터웨어하우스 패키지
Facebook에서 매일 같이 생산되는 대량의
데이터를 관리하고 학습하기 위해 개발

What is the data warehouse?
In computing, a data warehouse or enterprise data warehouse (DW, DWH, or EDW) is a database used for reporting and data analysis. It is a central repository of data which is created by integrating data from one or more disparate sources. Data warehouses store current as well as historical data and are used for creating trending reports for senior management reporting such as annual and quarterly comparisons

wikipidea(http://en.wikipedia.org/wiki/Data_warehouse)

하이브 설치하기

- 요구사항
  - Java 1.6 이상
  - Hadoop 0.20.x (버전은 크게 상관 없음)
- 설치 및 실행
  - http://mirror.apache-kr.org/hive/stable/hive-0.10.0.tar.gz 다운 로드 후 압축 해제
  - 환경 변수 설정: $HADOOP_HOME
  - 실행을 위한 기본 파일 생성 및 권한 설정
    - hadoop fs -mkdir /tmp
    - hadoop fs -chmod a+w /tmp
    - hadoop fs -mkdir /user/hive/warehouse
    - hadoop fs -chmod a+w /user/hive/warehouse
  - 하이브 셸 실행: hive

Hive 실행시 아래와 같은 예외가 발생하는 경우

FAILED: Error in metadata: java.lang.RuntimeException: Unable to instantiate org.apache.hadoop.hive.metastore.HiveMetaStoreClient
FAILED: Execution Error, return code 1 from org.apache.hadoop.hive.ql.exec.DDLTask

- 원인: hadoop을 직접 Build를 수행한 적이 있을 경우 classpath에 동일한 jar 파일이 여러번 참조되어 발생함

- 해결책: $HADOOP_HOME/build 디렉토리 삭제 or 이름 변경

하이브의 컨포넌트

하이브 설정

- config 파일 설정
  - hive-site.xml 파일이 아닌 경로를 설정
  - 하이브 셸 실행 시
    - hive --config <디렉토리 경로>
- 속성 설정
  - 하이브 셸 실행 시
    - hive -hiveconf fs.default.name=localhost
  - 셸 내에서
    - hive> SET hive.enforce.bucketing=true

속성 지정 우선 순위

1. 하이브 SET 명령어
2. 명령행 -hiveconf 옵션
3. hive-site.xml
4. hive-default.xml
5. hadoop-site.xml
6. hadoop-default.xml

로깅

- 설정
  - conf/hive-log4j.properties
- 기본 로그 파일 위치
  - /tmp/$USER/hive.log
- 하이브 셸 실행시 옵션 변경
  - hive -hiveconf hive.root.logger=DEBUG,console

하이브 서비스

cli

하이브 셸에 대한 명령행 인터페이스

hiveserver

쓰리프트, JDBC, ODBC 연결자를 사용하는 응용 프로그램은 하이브와 통신하기 위하여 하이브 서버를 필요

hwi

하이브 웹 인터페이스

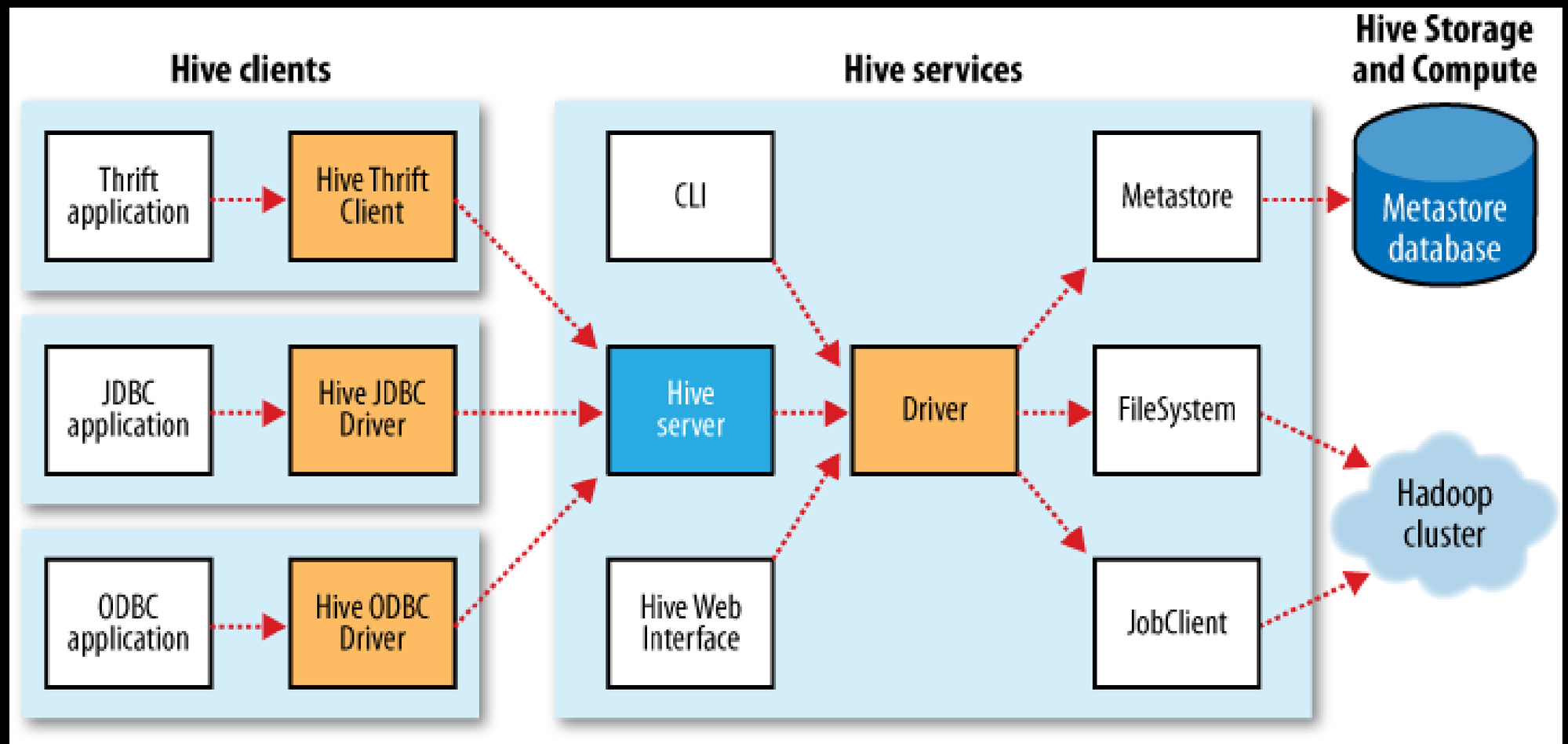
jar

classpath에 하둡과 하이브 클래스 모두를 사용하는 자바 응용 프로그램을 실행하기 위한 방법. hadoop jar 와

metastore

하이브 실행시 메타 데이터가 저장되는 장소

하이브 서비스 아키텍처

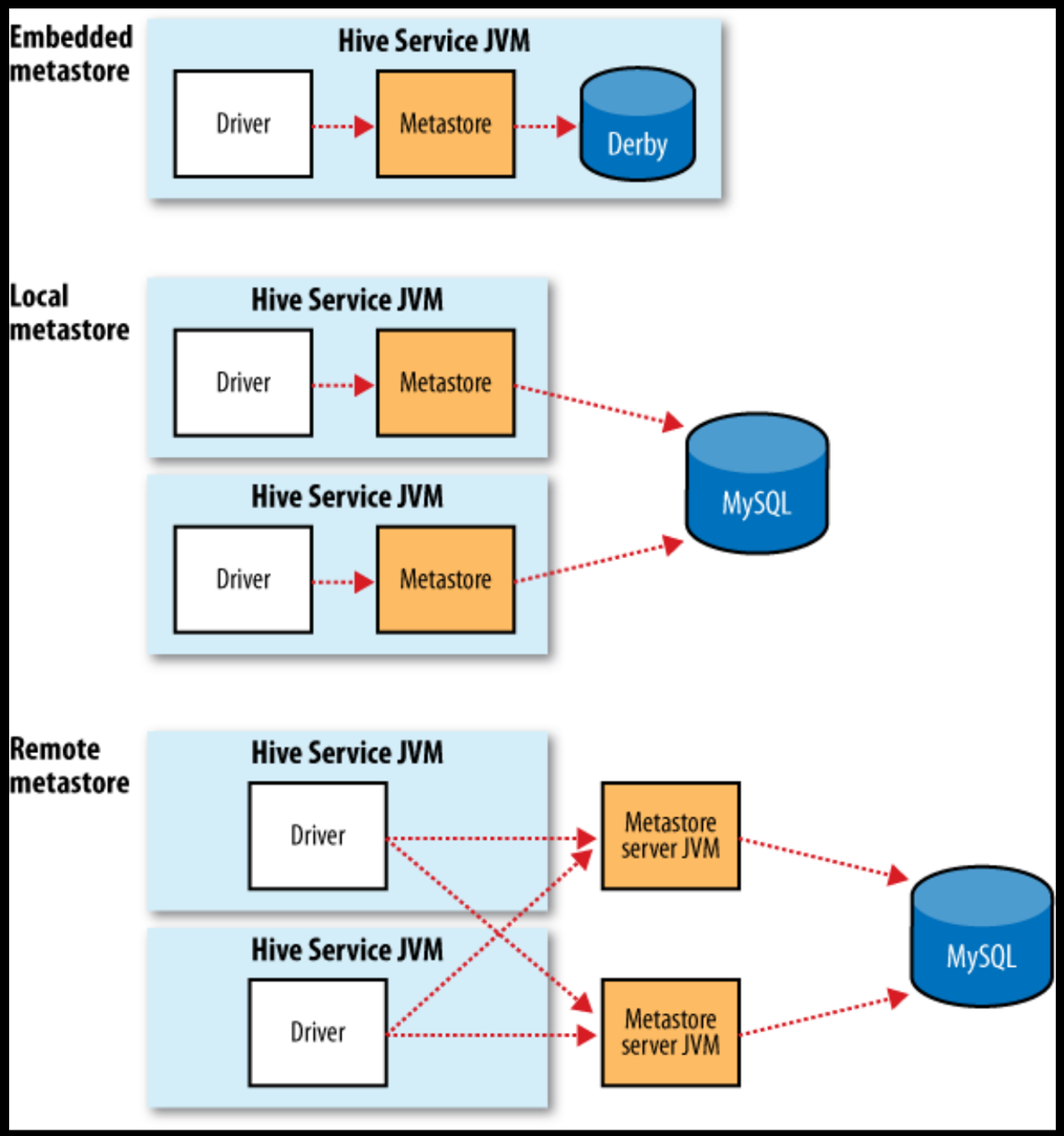

Hive clients
Hive services
Hive Storage and Compute
Thrift application
Hive Thrift Client
JDBC application
Hive JDBC Driver
ODBC application
Hive ODBC Driver
CLI
Hive server
Hive Web Interface
Driver
Metastore
FileSystem
JobClient
Metastore database
Hadoop cluster

메타 스토어

하이브 메타데이터에서 가장 중요한 저장소

- 서비스와 데이터 저장소로 이루어짐
- 기본 설정값
  - 메타 데이터 서비스: 하이브 서비스와 동일 JVM에서 실행
  - 데이터 저장소: 로컬 디스크 임베디드 더비 데이터 베이스

메타 스토어 설정



주요 속성

- 하이브 기본 디렉토리
  - hive.metastore.warehouse.dir
- 임베디드 메타 스토어 사용유무
  - hive.metastore.local
  - 0.10 버전에서 duplicate 됨
- 연결할 메타 스토어 서버
  - hive.metastore.uris
- ETC
  - javax.jdo.option.xxxx

전통적인 데이터 베이스와의 비교

테이블 스키마 검증 시점

전통적인 데이터 베이스

- 데이터를 적재하는 시점에 검증
- 만일 Insert중인 데이터가 스키마에 부합되지 않으면 데이터 거부
- 컬럼 단위로 색인이 가능하기 때문에 빠른 쿼리 성능을 제공

하이브

- 쿼리 실행시 데이터 검증
- 데이터의 매우 빠른 적재를 제공 (ex: 파일 복사 or 이동)
- 동일 데이터를 두 스키마로 다루어야 할때 훌륭한 유연성을 제공

갱신, 트랜잭션, 색인

- 하이브는 갱신을 지원하지 않음
- 락 매카니즘
  - 주키퍼를 사용하여 테이블과 파티션 수준의 락을 지원(0.7.0)
- 지원 색인
  - 0.7.0부터 특별한 경우를 위한 색인 기능을 제공
  - 컴팩트, 비트맵 타입 지원

HiveQL

표준 SQL과의 주요 차이

속성	SQL	HiveQL
갱신	UPDATE, INSERT, DELETE	INSERT OVERWRITE TABLE
데이터형	정수형, 부동 소수점, 고정소수점, 텍스트, 바이너리, 시간	추가적으로 배열, 맵, 구조체 지원
다중 테이블 삽입	지원하지 않음	지원
조인	FROM 절에서 테이블 조인 WHERE 절에서도 조건 지원 (SQL-92 지원)	내부조인, 외부조인, 세미조인, 맵조인지원 (WHERE절 조건 지원 X)
서브쿼리	어떠한 절에서도 사용가능	FROM 절에서만 사용 가능
뷰	업데이트 가능	읽기 전용
확장 가능성	사용자 정의 함수, 저장 프로시저	사용자 정의 함수, 맵-리듀스 스크립트

데이터형

기본형과 복합형 모두 지원

Category	Type	Description	Literal examples
Primitive	TINYINT	1-byte (8-bit) signed integer, from -128 to 127	1
	SMALLINT	2-byte (16-bit) signed integer, from -32,768 to 32,767	1
	INT	4-byte (32-bit) signed integer, from -2,147,483,648 to 2,147,483,647	1
	BIGINT	8-byte (64-bit) signed integer, from -9,223,372,036,854,775,808 to 9,223,372,036,854,775,807	1
	FLOAT	4-byte (32-bit) single-precision floating-point number	1.0
	DOUBLE	8-byte (64-bit) double-precision floating-point number	1.0
	BOOLEAN	true/false value	TRUE
	STRING	Character string	'a', "a"
	BINARY	Byte array	Not supported
	TIMESTAMP	Timestamp with nanosecond precision	1325502245000, '2012-01-02 03:04:05.123456789'

Category	Type	Description	Literal examples
Complex	ARRAY	An ordered collection of fields. The fields must all be of the same type.	array(1, 2)[a]
	MAP	An unordered collection of key-value pairs. Keys must be primitives; values may be any type. For a particular map, the keys must be the same type, and the values must be the same type.	map('a', 1, 'b', 2)
	STRUCT	A collection of named fields. The fields may be of different types.	struct('a', 1, 1.0)[b]

연산자와 함수

- 관계형, 산술, 논리 연산자와 같은 일반적인 SQL 연산자 제공
  - "||"는 문자열 연결이 아니라 논리 OR
- 여러가지 내장 함수가 포함되어 있음
  - hive> SHOW FUNCTIONS
- 형 변환
  - 암묵적 형변환을 수행하기 위해 기본형에 대한 계층 구조를 가짐
  - 암묵적 원칙
    - 정수형, FLOAT, STRING => DOUBLE
    - TINY INT, SMALL INT, INT => FLOAT
    - Boolean은 어떠한 형으로 변환 안됨
  - 명시적 형변환: CAST 사용

테이블

하이브 테이블은 논리적으로 저장된 데이터와
테이블에서 데이터의 배치를 기술하는 메타데이터로 구성

관리 테이블과 외부 테이블

- 하이브는 기본적으로 테이블을 직접 관리
  - 데이터를 데이터웨어하우스로 이동한다는 의미
- 외부 테이블
  - 웨어하우스 디렉토리 외부에서 데이터를 참조
- 차이점
  - 삭제시 외부 테이블은 데이터를 삭제하지 않음

파티션

컬럼의 값을 기반으로 크게 나눠놓는 방식
관련된 파티션의 파일만 스캔하면 되기 때문에 효율적

EX) 날짜와 국가 코드로
파티셔닝할 경우

```
CREATE TABLE logs (ts BIGINT, line STRING)
PARTITIONED BY (dt STRING, country STRING);

LOAD DATA LOCAL INPATH 'input/hive/partitions/file1'

INTO TABLE logs PARTITION (dt='2001-01-01', country='GB');
```

저장 디렉토리

```
/user/hive/warehouse/logs/dt=2010-01-01/country=GB/file1
                                              /file2
                         /country=US/file3
                /dt=2010-01-02/country=GB/file4
                         /country=US/file5 /file6
```

버킷

- 테이블이나 파티션에서 더 효율적 쿼리를 위해 사용
- 사용 이유
  - 추가 구조를 부여하여 효율적인 쿼리가 가능(조인, 정렬 등)
  - 샘플링을 효율적으로 할수 있음
- 버킷은 테이블 디렉토리 안에 파일 하나로 표현

```
CREATE TABLE bucketed_users (id INT, name STRING)
CLUSTERED BY (id) INTO 4 BUCKETS;

CREATE TABLE bucketed_users (id INT, name STRING)
CLUSTERED BY (id) SORTED BY (id ASC) INTO 4 BUCKETS;
// 정렬, 외부에서 생성된 데이터를 버킷팅된 테이블로 적재가 가능하지만
일반적으로는 존재하는 테이블에서 버킷팅 수행
```

```
hive> SELECT * FROM users;
0 Nat
2 Joe
3 Kay
4 Ann

INSERT OVERWRITE TABLE bucketed_users
SELECT * FROM users;
```

기타 테이블 연산

- 데이터 임포트하기
  - INSERT or CATS(CREATE TABLE .. AS .. SELECT)
  - ex) INSERT OVERWRITE TABLE target SELECT col1, col2 FROM source;
- 테이블 변경하기
  - ALTER TABLE
  - ex) ALTER TABLE target ADD COLUMNS (col3 STRING);
- 테이블 삭제하기
  - DROP TABLE
  - 데이터만 지울 경우 warehouse 디렉토리 내의 파일 제거

데이터 쿼리하기

정렬과 집계

ORDER BY

- 전체적으로 정렬된 결과를 생성, 이를 위해 리듀서를 하나로 설정해야 함
- 대규모 데이터셋에서 매우 비효율적

SORT BY

- 리듀서당 정렬된 파일을 생성

DISTRIBUTE BY

- 특정 로우가 특정 리듀서로 가도록 보장

```
FROM records2
SELECT year, temperature
DISTRIBUTE BY year
SORT BY year ASC, temperature DESC;
```

맵 리듀스 스크립트

하둡 스트리밍과 같은 방식을 사용하여
TRANSFORM, MAP, REDUCE 절을
통해 외부 스크립트나 프로그램을 호출 가능

```
#!/usr/bin/env python
import re import sys
for line in sys.stdin:
(year, temp, q) = line.strip().split()
if (temp != "9999" and re.match("[01459]", q)):
print "%s\t%s" % (year, temp)
```

```
ADD FILE /path/to/is_good_quality.py;

FROM records2
SELECT TRANSFORM(year, temperature, quality)
USING 'is_good_quality.py'
AS year, temperature;
```

조인

하이브는 내부조인, 외부조인, 세미조인, 맵 조인을 지원

- 맵조인: 테이블 하나가 메모리에 적재될 만큼 충분히 작다면 하이브는 매퍼 단위로 조인을 수행할 수 있도록 지원

EX) 내부 조인 EX

```
hive> SELECT * FROM sales;
Joe 2
Hank 4
Ali 0
Eve 3
Hank 2

hive> SELECT * FROM things;
2 Tie
4 Coat
3 Hat
1 Scarf
```

```
hive> SELECT sales.*, things.*
> FROM sales JOIN things ON (sales.id = things.id);
Joe     2   2   Tie
Hank    2   2   Tie
Eve     3   3   Hat
Hank    4   4   Coat
```

서브 쿼리 & 뷰

```
SELECT station, year, AVG(max_temperature)
FROM (
      SELECT station, year, MAX(temperature) AS
      max_temperature FROM records2
      WHERE temperature != 9999
      AND (quality = 0 OR quality = 1 OR quality = 4
      OR quality = 5 OR quality = 9) GROUP BY
      station, year
      ) mt
GROUP BY station, year;
```

```
CREATE VIEW valid_records AS
SELECT *
FROM records2
WHERE temperature != 9999
      AND (quality = 0 OR quality = 1 OR quality = 4
      OR quality = 5 OR quality = 9);
```

References

1. Tom White (2013). 하둡 완벽가이드. (심탁길, 김현우, 옮김). 서울: 한빛미디어. (원서출판 2012)