

학습목표

1. 샤딩의 개념
2. 스파이더 샤딩 플랫폼
3. 도커기반 샤딩실습

학습내용

데이터베이스 샤딩에 대해 알 수 있습니다.

사전퀴즈

1. Spider 기반 샤딩을 하려면 별도의 엔진을 추가해야 한다.
정답 : O
해설 : Spider 샤딩 플랫폼은 Spider 엔진을 추가해서 샤딩을 구현한다.
2. 샤딩을 지속적으로 구현하면 전체적으로 쓰기 성능이 올라간다.
정답 : O
해설 : 샤딩을 구축하면 읽기성능보다는 쓰기 성능이 향상된다.

수업

1. 샤딩

VLDB(Very Large DBMS)

- 전체 DB가 하나의 DBMS 시스템에 다 들어가기 힘들어지는 경우
- 테이블들을 여러 개의 군으로 나눠 분산 저장
예) 테이블 1,2는 서버 1에 테이블 3,4는 서버 2에 저장
 - 하나의 테이블이 방대한 경우에는 사전방식 (a-m, n-r, s-z)와 같이 나눠 저장
(샤딩 플랫폼에서 제공하는 샤딩 방식들을 사용)

파티셔닝(Partitioning)

DBMS 레벨 분할

샤딩(Sharding)

- DBMS 외부에서 분할
- 응용레벨에서 구별해야 함

샤딩과 파티셔닝의 차이점?

<http://theeye.pe.kr/archives/1917>

수평(Horizontal)/수직(Vertical) 샤딩

수평(Horizontal)	수직(Vertical)
동일한 시스템을 지역/국가별로 나눠서 구성하는 경우	샤드를 만들어 저장할 때 구분 기준을 연령/성별과 같은 기준으로 나눠 구

	성
동일한 테이블을 여러개로 분할	테이블 단위로 분할

데이터베이스 파티셔닝(Partitioning)

파티셔닝 방식에 준함 (샤딩 플랫폼이 지원하는 방식 확인)

- 범위(range) : a-m / n-r / s-z
- 해시(hash) : 해시함수 파티션별로 크기를 비슷하게 나눔
- 리스트(list) : 특정한 컬럼을 기준
- 콤포지트(composite) : range-hash / range-list

스파이더(Spider) 샤딩 플랫폼

- MariaDB/MySQL용 샤딩 플랫폼
 - MariaDB에 자체 내장됨
 - 현재 10.1까지만 지원
- 최소 3개의 MariaDB 인스턴스 필요
 - Spider 인스턴스/샤딩 인스턴스 2개이상
 - Spider 인스턴스 : 어느 샤드에 들어갈지 판단해서 저장해주는 인스턴스
 - 샤딩 인스턴스 : 실제 저장하는 인스턴스

실습

- 1) mariadb 다운로드
 docker pull mariadb:10.1
- 2) spider, sample1, sample2 인스턴스 실행

```

MINGW64 ~
$ docker run -d -e MYSQL_ROOT_PASSWORD=sample --name=spider mariadb:10.1
9bbd6eea91d1144d522a07c39a909ab6f4a7b9228e2f6eb4db0d3b3ddd878b3e

MINGW64 ~
$ docker run -d -e MYSQL_ROOT_PASSWORD=sample --name=sample1 mariadb:10.1
7dcbd6e77c3a068383120cdf5d88d574338704ed8356c6a9f635301ea7b54cbf

MINGW64 ~
$ docker run -d -e MYSQL_ROOT_PASSWORD=sample --name=sample2 mariadb:10.1
8c3c0bbeacbc155d2018c9903c70ec2e1etdf294bebucd1a196eb77eba7c1be8

MINGW64 ~
$ docker ps -a

```

CONTAINER ID	IMAGE	COMMAND	CREATED	STATUS	PORTS	NAMES
8c3c0bbeacbc	mariadb:10.1	"docker-entrypoint.s..."	7 seconds ago	Up 5 seconds	3306/tcp	sample2
7dcbd6e77c3a	mariadb:10.1	"docker-entrypoint.s..."	13 seconds ago	Up 11 seconds	3306/tcp	sample1
9bbd6eea91d1	mariadb:10.1	"docker-entrypoint.s..."	28 seconds ago	Up 27 seconds	3306/tcp	spider

- 3) spider, sample1, sample2 인스턴스에서 IP주소 확인

```

MINGW64 ~
$ docker inspect spider | grep "IPAddress"
    "SecondaryIPAddresses": null,
    "IPAddress": "172.17.0.2",
    "IPAddress": "172.17.0.2",

MINGW64 ~
$ docker inspect sample1 | grep "IPAddress"
    "SecondaryIPAddresses": null,
    "IPAddress": "172.17.0.3",
    "IPAddress": "172.17.0.3",

MINGW64 ~
$ docker inspect sample2 | grep "IPAddress"
    "SecondaryIPAddresses": null,
    "IPAddress": "172.17.0.6",
    "IPAddress": "172.17.0.6",

```

4) spider 인스턴스 접속 후 스파이더 설치, 스파이더 엔진 설치 확인

```

MINGW64 ~
$ docker exec -it spider bash
root@9bbd6eea91d1:/# cd /usr/share/mysql
root@9bbd6eea91d1:/usr/share/mysql# ls -al
total 1252
drwxr-xr-x 27 root root 4096 Nov 19 22:28 .
drwxr-xr-x 1 root root 4096 Nov 19 22:28 ..
drwxr-xr-x 2 root root 4096 Nov 19 22:28 charsets
drwxr-xr-x 2 root root 4096 Nov 19 22:28 czech
drwxr-xr-x 2 root root 4096 Nov 19 22:28 danish
-rwxr-xr-x 1 root root 2671 Oct 31 21:58 debian-start.inc.sh
drwxr-xr-x 2 root root 4096 Nov 19 22:28 dutch
-rwxr-xr-x 1 root root 27 Oct 31 21:58 echo_stderr
drwxr-xr-x 2 root root 4096 Nov 19 22:28 english
-rw-r--r-- 1 root root 501046 Oct 31 21:54 errmsg-utf8.txt
drwxr-xr-x 2 root root 4096 Nov 19 22:28 estonian
-rw-r--r-- 1 root root 583827 Oct 31 21:54 fill_help_tables.sql
drwxr-xr-x 2 root root 4096 Nov 19 22:28 french
drwxr-xr-x 2 root root 4096 Nov 19 22:28 german
drwxr-xr-x 2 root root 4096 Nov 19 22:28 greek
drwxr-xr-x 2 root root 4096 Nov 19 22:28 hungarian
-rw-r--r-- 1 root root 14335 Oct 31 21:54 install_spider.sql

root@9bbd6eea91d1:/usr/share/mysql# mysql -uroot -p < install_spider.sql
Enter password:
root@9bbd6eea91d1:/usr/share/mysql# mysql -uroot -p
Enter password:
Welcome to the MariaDB monitor.  Commands end with ; or \g.
Your MariaDB connection id is 3
Server version: 10.1.37-MariaDB-1-bionic mariadb.org binary distribution
Copyright (c) 2000, 2018, Oracle, MariaDB Corporation Ab and others.

Type 'help;' or '\h' for help. Type '\c' to clear the current input statement.

MariaDB [(none)]> show engines;

```

Engine	Support	Comment	Transactions	XA	Savepoints
SPIDER	YES	Spider storage engine	YES	YES	NO
MyISAM	YES	Collection of identical MyISAM tables	NO	NO	NO
CSV	YES	Stores tables as CSV files	NO	NO	NO
MEMORY	YES	Hash based, stored in memory, useful for temporary tables	NO	NO	NO
MyISAM	YES	Non-transactional engine with good performance and small data footprint	NO	NO	NO
SEQUENCE	YES	Generated tables filled with sequential values	YES	NO	YES
Aria	YES	Crash-safe tables with MyISAM heritage	NO	NO	NO
PERFORMANCE_SCHEMA	YES	Performance Schema	NO	NO	NO
InnoDB	DEFAULT	Percona-XtraDB, Supports transactions, row-level locking, foreign keys and encryption for tables	YES	YES	YES

5) spider, sample1, sample2 인스턴스에서 sampleDB 데이터베이스 생성, 사용자추가 및 권한할당

```

MariaDB [(none)]> create database sampleDB;
Query OK, 1 row affected (0.00 sec)

MariaDB [(none)]> create user 'spider-test'@'%' identified by 'sample';
Query OK, 0 rows affected (0.00 sec)

MariaDB [(none)]> grant all on *.* to 'spider-test'@'%' with grant option;
Query OK, 0 rows affected (0.00 sec)

MariaDB [(none)]> flush privileges;
Query OK, 0 rows affected (0.00 sec)

```

6) spider 인스턴스에서 sample1, sample2 서버정보 등록하기

```
MariaDB [(none)]> CREATE SERVER sample1
-> FOREIGN DATA WRAPPER mysql
-> OPTIONS(
-> HOST '172.17.0.3',
-> DATABASE 'sampleDB',
-> USER 'spider-test',
-> PASSWORD 'sample',
-> PORT 3306
-> )
-> ;
Query OK, 0 rows affected (0.00 sec)

MariaDB [(none)]> CREATE SERVER sample2
-> FOREIGN DATA WRAPPER mysql
-> OPTIONS(
-> HOST '172.17.0.6',
-> DATABASE 'sampleDB',
-> USER 'spider-test',
-> PASSWORD 'sample',
-> PORT 3306
-> )
-> ;
Query OK, 0 rows affected (0.00 sec)

MariaDB [(none)]> select * from mysql.servers#G;
***** 1. row *****
Server_name: sample1
Host: 172.17.0.3
Db: sampleDB
Username: spider-test
Password: sample
Port: 3306
Socket:
Wrapper: mysql
Owner:
***** 2. row *****
Server_name: sample2
Host: 172.17.0.6
Db: sampleDB
Username: spider-test
Password: sample
Port: 3306
Socket:
Wrapper: mysql
Owner:
2 rows in set (0.00 sec)
```

7) Spider 인스턴스에서 테이블 생성

- spider 엔진을 가지는 shardingTest 테이블
- COMMENT 부분을 통해서 spider 플랫폼과 통신하게 된다
- shardingTest 테이블을 파티션한 것처럼 보이지만 실제로는 샤드로 돌아가는 것임
(sample1 파티션, sample2 파티션은 실제로 각각 sample1, sample2 인스턴스를 의미한다)

```

MariaDB [(none)]> use sampleDB;
Database changed
MariaDB [sampleDB]> CREATE TABLE shardingTest(
  -> id int not null auto_increment,
  -> name varchar(255) not null,
  -> address varchar(255) not null,
  -> primary key(id)
  -> ) ENGINE=spider COMMENT='wrapper "mysql", table "shardingTest"'
  -> PARTITION BY KEY(id)(
  -> PARTITION sample1 COMMENT='srv "sample1"',
  -> PARTITION sample2 COMMENT='srv "sample2"'
  -> )
  -> ;
Query OK, 0 rows affected (0.03 sec)

MariaDB [sampleDB]> show create table shardingTest\G;
***** 1. row *****
      Table: shardingTest
Create Table: CREATE TABLE `shardingTest` (
  `id` int(11) NOT NULL AUTO_INCREMENT,
  `name` varchar(255) NOT NULL,
  `address` varchar(255) NOT NULL,
  PRIMARY KEY (`id`)
) ENGINE=SPIDER DEFAULT CHARSET=latin1 COMMENT='wrapper "mysql", table "shardingTest"'
/*!50100 PARTITION BY KEY (`id`)
(PARTITION sample1 COMMENT = 'srv "sample1"' ENGINE = SPIDER,
PARTITION sample2 COMMENT = 'srv "sample2"' ENGINE = SPIDER) */
1 row in set, 2 warnings (0.01 sec)

```

8) Sample1,2 인스턴스에서 테이블 생성

```

MariaDB [(none)]> use sampleDB;
Database changed
MariaDB [sampleDB]> CREATE TABLE shardingTest(
  -> id int not null auto_increment,
  -> name varchar(255) not null,
  -> address varchar(255) not null,
  -> primary key(id)
  -> ) ENGINE=innodb
  -> ;
Query OK, 0 rows affected (0.10 sec)

MariaDB [sampleDB]> show create table shardingTest\G;
***** 1. row *****
      Table: shardingTest
Create Table: CREATE TABLE `shardingTest` (
  `id` int(11) NOT NULL AUTO_INCREMENT,
  `name` varchar(255) NOT NULL,
  `address` varchar(255) NOT NULL,
  PRIMARY KEY (`id`)
) ENGINE=InnoDB DEFAULT CHARSET=latin1
1 row in set (0.00 sec)

```

9) Spider 인스턴스에서 레코드 생성

```

MariaDB [sampleDB]> insert into shardingTest(name, address) values('kim', 'seoul');
Query OK, 1 row affected (0.05 sec)

MariaDB [sampleDB]> insert into shardingTest(name, address) values('lee', 'seoul');
Query OK, 1 row affected (0.01 sec)

MariaDB [sampleDB]> insert into shardingTest(name, address) values('park', 'seoul');
Query OK, 1 row affected (0.02 sec)

MariaDB [sampleDB]> insert into shardingTest(name, address) values('kim', 'busan');
Query OK, 1 row affected (0.02 sec)

MariaDB [sampleDB]> insert into shardingTest(name, address) values('lee', 'daegu');
Query OK, 1 row affected (0.02 sec)

MariaDB [sampleDB]> insert into shardingTest(name, address) values('park', 'jeju');
Query OK, 1 row affected (0.01 sec)

MariaDB [sampleDB]> select * from shardingTest;
+----+-----+-----+
| id | name | address |
+----+-----+-----+
| 1  | kim  | seoul   |
| 3  | park | seoul   |
| 5  | lee  | daegu   |
| 2  | lee  | seoul   |
| 4  | kim  | busan   |
| 6  | park | jeju    |
+----+-----+-----+
6 rows in set (0.00 sec)

```

10) sample1, sample2인스턴스에서 shardingTest 테이블 확인

sample1

```

MariaDB [sampleDB]> select * from shardingTest;
+----+-----+-----+
| id | name | address |
+----+-----+-----+
| 1  | kim  | seoul   |
| 3  | park | seoul   |
| 5  | lee  | daegu   |
+----+-----+-----+
3 rows in set (0.00 sec)

```

sample2

```

MariaDB [sampleDB]> select * from shardingTest;
+----+-----+-----+
| id | name | address |
+----+-----+-----+
| 2  | lee  | seoul   |
| 4  | kim  | busan   |
| 6  | park | jeju    |
+----+-----+-----+
3 rows in set (0.00 sec)

```

샤딩과 파티셔닝의 차이

전문가의견

샤딩	파티셔닝
DBMS 외부의 플랫폼에서 지원하는 기능이라 개념 샤드 별로 상태를 조정할 수 있다.	DBMS 자체에서 제공하는 개념이라 사용자는 개별 파티션의 상태를 조회할 수는 없다.

물리적인 시스템으로 분할해서 관리한다.	관리할 대상을 나눠서 읽기/쓰기 최적화한다.
지속적으로 쓰기 성능이 향상	성능개선이 어느정도까지만 되고 파티션을 많이 만들면 오히려 성능이 떨어지는 경우도 있다.