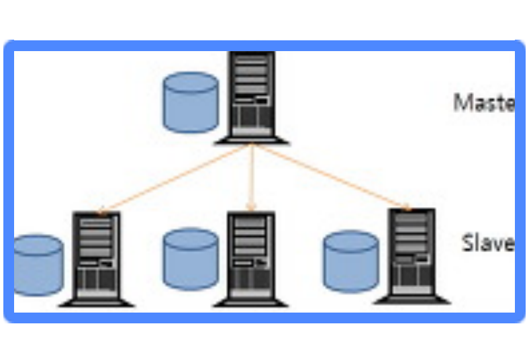
MySQL Replication 설정과 몇 가지 테스트

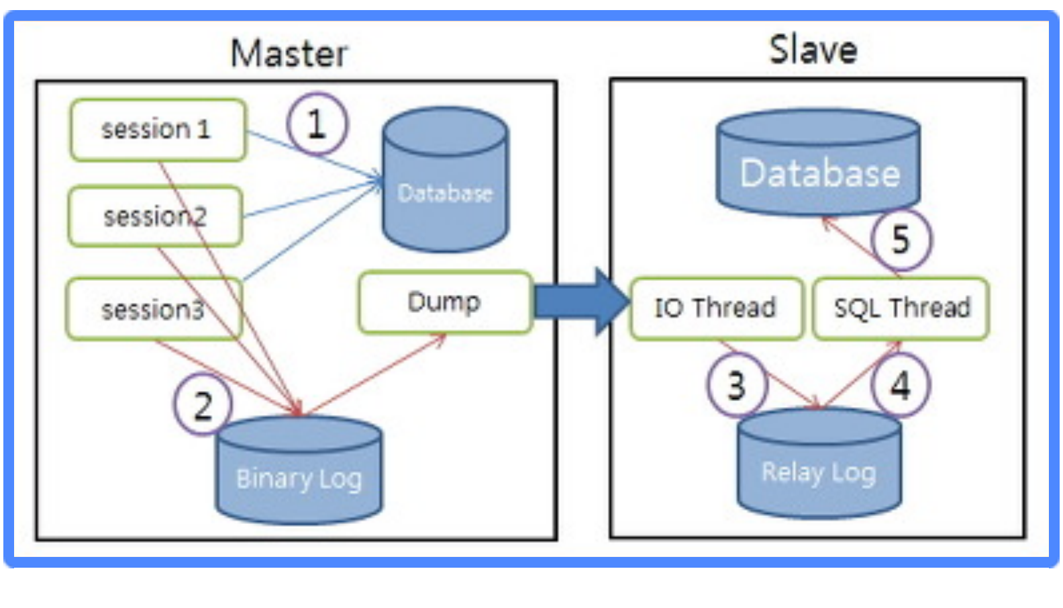
2013/02/07 15:17



MySQL을 DB로 사용하면서 서버의 부하 분산을 위한 방법 중 하나로 Replication 을 사용한다.
Replication 은 Master 하나에 n개의 Slave로 지정이 가능하다. Slave는 다시 Master 역할을 할수 있으며 역시 또 다른 n개의 Slave를 지정할 수 있다.
부하 분산의 효과는 inser,update 등 변경과 관련된 모든 작업은 Master에서 실행을 하고, select 등 읽기과 관련된 작업은 Slave를 통하여 작업을 함으로서 읽기와 쓰기에 대한 부하분산이 가능하다.

MySQL의 Replication 은 비동기 방식으로 처리 되며, Master 에서 변경된 내용이 Slave에 적용이 된다.

MySQL 에서 Replication 이 적용되는 방식은 아래 그림을 요약할 수 있다.



1. Master 에서 데이터의 변경작업이 발생하면 Master DB의 변경 실행
2. 변경된 이력을 Binary Log 에 기록
3. Slave 의 IO Thread 가 Master 의 변경 Event 를 감지하고 Master 의 Binary Log 를 자신(Slave)의 Relay Log 에 기록한다.
4. Slave 의 SQL Thread 는 Relay Log 를 읽고 자신(Slave)의 DB에 기록한다.

위 그림에서처럼 Master 에서 Slave 방향으로 단방향으로만 처리가 이루어 지므로 변경작업(Write)는 Master에만 하여야 한다.
Slave에서 변경작업을 할 경우 Master에는 당연히 반영이 안되며, Slave에서 변경작업한 Object 에 대하여 Master 서 변경작업을 다시 시도할 경우 변경은 되지만 다시 slave에 는 적용되지 않는다.

MySQL Replication 설정

1. 기본적으로 Master와 Slave로 사용할 Mysql 서버가 각각 설치되어 있어야 한다. 가능하면 물리적으로 다른 서버가 필요하지만 port 를 달리 설치한 물리적 동일 서버내에서 테스트하는것도 가능할것 같다.
Replication 기능은 3.23.15부터 지원되기 시작하였으나, 3.23.32부터 안정화 되었기에 가능하면 최신버전의 MySQL을 권한다.(현재 5.6 버전이 release 되었는데......)

설정 요약
-. Master 1개에 Slave 1개를 지정
-. Master
    IP: 192.168.0.1
    server-id = 1
    replication 대상 db명 : classmate
    계정 : root('rootpassword')
    replication 권한 계정 : rep('reppassword')
-. Slave
    IP : 192.168.0.2
    server-id = 2

1. Master 계정생성
-. Slave 에서 Master 로 접근 할수 있는 계정이 필요하다. 이 계정은 'REPLICATION SALVE' 권한이 기본적으로 필요하다
Master 서버에서 아래와 같이 권한을 지정한다.

```
master mysql > GRANT REPLICATION SLAVE ON *.* TO 'rep'@'192.168.0.2' IDENTIFIED BY
'reppassword';
```

2. Master Data 를 Slave로 복사
-. HOT 백업

```
master mysql > FLUSH TABLES WITH READ LOCK;
master shell > tar -cvf /tmp/mysql-snapshot.tar .
slave   shell > tar -xvf /tmp/mysql-snapshot.tar
master mysql > UNLOCK TABLES;
```

-. mysqldump 이용 백업

```
master Shell > mysqldump -u root -p 'rootpassword'  classmate > dump_classmate.sql       db 명 :
classmate
--아래 slave 서버로의 dump 파일을 이동 & load 는 '6'에서 실행하면 됨--
dump_classmate.sql 파일을 slave 로 복사 후
slave Shell > mysql -u root -p 'rootpassword'  classmate < dump_classmate.sql       db 명 :
classmate
```

3. Master 환경설정
-. master의 my.cnf 파일을 수정한다.

```
master shell> vi /etc/my.cnf

[mysqld]
log-bin=mysql-bin
server-id  = 1
```

-. master 의 server-id 를 1로 지정한다.(다른숫자 값으로 하여도 상관없다. slave 와 중복되지 않으면 된다(1~(2^32)-1내의 숫자)
-. [mysqld]섹션에 위와 같이 log-bin 이 존재하는지 확인한다

4. Master 의 Replication 정보 조회

```
master mysql> SHOW MASTER STATUS;
+------------------+----------+--------------+------------------+
| File | Position | Binlog_Do_DB | Binlog_Ignore_DB |
+------------------+----------+--------------+------------------+
| mysql-bin.000001 | 98 | | |
+------------------+----------+--------------+------------------+
```

-. File : 로그 파일
-. Position : 로그 파일내 읽을 위치
-. Binlog_Do_DB : binary log 파일(변경된 이벤트 정보가 쌓이는 파일)
-. Binlog_Ignore_DB : 복제 제외 정보
    n Binlog_Do_DB와 Binlog_Ignore_DB는 Slave 시작하기 전까지는 나타나지 않는다.
-. File 과 Position 정보가 '7'의 과정에서 필요하다.

5. Slave 환경 설정
-. slave 의 my.cnf 파일을 수정한다.

```
slave shell> vi /etc/my.cnf



[mysqld]
server-id  = 2
master-host = 192.168.0.1
master-port = 3306
master-user = rep (위에서 설정한 replication을 위한 계정 정보)
master-password = reppassword
```

6. Slave 서버 Start
-. 2의 과정에서 HOT 백업을 하였다면 Slave 서버의 mysql data 디렉토리에 master 서버의 데이터를 복사하고, mysqldump 를 이용하여 백업하였다면, mysql 서버를 start 후 dump 파일을 load 한다.

```


slave shell > /etc/init.d/mysqld start

2에서 mysldump 로 dump 한 dump_classmate.sql 파일을 slave 로 복사 후
slave Shell > mysql -u root -p 'rootpassword' classmate < dump_classmate.sql   (db 명 : classmate)
master에서 classamte db를 dump 한 dump_classmate.sql 파일을 이용하여 slave의 clsssmate db로
load(slave에는 classmate 라는 db가 존재하여야 함)
```

7. Slave 정보 설정
-. Slave 에서 Master로 연결하기 위한 정보를 설정한다.

```
slave mysql > CHANGE MASTER TO
MASTER_HOST='192.168.0.1',
MASTER_USER='rep',
MASTER_PASSWORD='reppassword',
MASTER_LOG_FILE='mysql-bin.000001',
MASTER_LOG_POS=98;
```

-. 4에서 조회한 master 의 logfile과 position 정보를 지정한다.

8. Slave Start

```
slave mysql > start slave;
```

이제 Master 변경이 되면 자동으로 Slave에서도 변경된 것을 확인 할 수 있다.

몇가지 Test 내용

-. 기본적인 master에서 데이터입력과 slave에서의 확인

```
1. master의 classmate DB의 tb_textbook 정보 조회
master mysql> select * from tb_textbook;
+-------------+------------+---------------+---------------+--------+-----------+--------------+---------+-----
------+
| textbook_id | subject_id | textbook_type | textbook_name | author | publisher | publish_year |
main_yn | refer_url |
+-------------+------------+---------------+---------------+--------+-----------+--------------+---------+-----
------+
|           7 |          1 |             1 | 1             | 1      | 1         |            1 | 1       | 1         |
|           6 |          1 |             1 | 1             | 1      | 1         |            1 | 1       | 2         |
+-------------+------------+---------------+---------------+--------+-----------+--------------+---------+-----
------+
2 rows in set (0.00 sec)

2. master 에서 tb_textbook에 1개의 row 등록
master mysql> INSERT INTO `classmatetest`.`tb_textbook` (`textbook_id`, `subject_id`,
`textbook_type`, `textbook_name`, `author`, `publisher`, `publish_year`, `main_yn`, `refer_url`) VALUES
('10', '2', '2', '교양 컴퓨터', '교수님', '출판사', '2013', 'n', 'http://hibrain.net');


mysql> select * from tb_textbook;
+-------------+------------+---------------+-----------------+-----------+-----------+--------------+---------
+--------------------+
| textbook_id | subject_id | textbook_type | textbook_name   | author    | publisher | publish_year |
main_yn | refer_url          |
+-------------+------------+---------------+-----------------+-----------+-----------+--------------+---------
+--------------------+
|           7 |          1 |             1 | 1               | 1         | 1         |            1 | 1       | 1                  |
|           6 |          1 |             1 | 1               | 1         | 1         |            1 | 1       | 2                  |
|          10 |          2 |             2 | 교양 컴퓨터     | 교수님    | 출판사    |         2013 | n       | http://hibrain.net |
+-------------+------------+---------------+-----------------+-----------+-----------+--------------+---------
+--------------------+
3 rows in set (0.00 sec)

slave에서 정보 조회
mysql> select * from tb_textbook;
+-------------+------------+---------------+-----------------+-----------+-----------+--------------+---------
+--------------------+
| textbook_id | subject_id | textbook_type | textbook_name   | author    | publisher | publish_year |
main_yn | refer_url          |
+-------------+------------+---------------+-----------------+-----------+-----------+--------------+---------
+--------------------+
|           7 |          1 |             1 | 1               | 1         | 1         |            1 | 1       | 1                  |
|           6 |          1 |             1 | 1               | 1         | 1         |            1 | 1       | 2                  |
|          10 |          2 |             2 | 교양 컴퓨터     | 교수님    | 출판사    |         2013 | n       | http://hibrain.net |
+-------------+------------+---------------+-----------------+-----------+-----------+--------------+---------
+--------------------+
3 rows in set (0.00 sec)
```

-. Master 에서의 Table 컬럼정보 변경

```
1. Master tb_textbook 정보

mysql> desc tb_textbook;
+---------------+--------------+------+-----+---------+-------+
| Field         | Type         | Null | Key | Default | Extra |
+---------------+--------------+------+-----+---------+-------+
| textbook_id   | int(11)      | NO   |     | NULL    |       |
| subject_id    | int(11)      | NO   |     | NULL    |       |
| textbook_type | int(11)      | NO   |     | NULL    |       |
| textbook_name | varchar(100) | NO   |     | NULL    |       |
| author        | varchar(100) | YES  |     | NULL    |       |
| publisher     | varchar(100) | YES  |     | NULL    |       |
| publish_year  | int(11)      | YES  |     | NULL    |       |
| main_yn       | char(1)      | YES  |     | NULL    |       |
| refer_url     | varchar(200) | YES  |     | NULL    |       |
+---------------+--------------+------+-----+---------+-------+
9 rows in set (0.00 sec)

2. Master 에서 tb_textbook의 refer_url 에 대하여 varchar(200) ==> int 로 변경
ALTER TABLE `tb_textbook` CHANGE `refer_url` `refer_url` INT NULL DEFAULT
NULL

3.Slave에서 확인
mysql> desc tb_textbook;
+---------------+--------------+------+-----+---------+-------+
| Field         | Type         | Null | Key | Default | Extra |
+---------------+--------------+------+-----+---------+-------+
| textbook_id   | int(11)      | NO   |     | NULL    |       |
| subject_id    | int(11)      | NO   |     | NULL    |       |
| textbook_type | int(11)      | NO   |     | NULL    |       |
| textbook_name | varchar(100) | NO   |     | NULL    |       |
| author        | varchar(100) | YES  |     | NULL    |       |
| publisher     | varchar(100) | YES  |     | NULL    |       |
| publish_year  | int(11)      | YES  |     | NULL    |       |
| main_yn       | char(1)      | YES  |     | NULL    |       |
| refer_url     | int(11)      | YES  |     | NULL    |       |
+---------------+--------------+------+-----+---------+-------+
9 rows in set (0.01 sec)
```

-. 컬럼 추가, 삭제의 과정도 마찬가지로 위의 과정과 동일하게 확인이 가능하다

-. Index 생성

```
master mysql> show index from tb_textbook;
+-------------+------------+-----------------------+--------------+-------------+-----------+-------------+--
--------+--------+------+------------+---------+---------------+
| Table       | Non_unique | Key_name              | Seq_in_index | Column_name | Collation | Cardinality |
Sub_part | Packed | Null | Index_type | Comment | Index_comment |
+-------------+------------+-----------------------+--------------+-------------+-----------+-------------+--
--------+--------+------+------------+---------+---------------+
| tb_textbook |          0 | PRIMARY               |            1 | textbook_id | A         |           3 |     NULL |
NULL   |      | BTREE      |         |               |
+------------+-----------+----------------------+-------------+------------+----------+------------+---------+--------+-
-----+-----------+---------+---------------+
1 rows in set (0.00 sec)
master mysql > ALTER TABLE `classmatetest`.`tb_textbook` ADD INDEX `textbook_subject_id_idx` ( `subject_id` );
 mysql> show index from tb_textbook;
+-------------+------------+-----------------------+--------------+-------------+-----------+-------------+--
--------+--------+------+------------+---------+---------------+
| Table       | Non_unique | Key_name              | Seq_in_index | Column_name | Collation | Cardinality |
Sub_part | Packed | Null | Index_type | Comment | Index_comment |
+-------------+------------+-----------------------+--------------+-------------+-----------+-------------+--
--------+--------+------+------------+---------+---------------+
| tb_textbook |          0 | PRIMARY               |            1 | textbook_id | A         |           3 |     NULL |
NULL   |      | BTREE      |         |               |
| tb_textbook |          1 | textbook_subject_id_idx |          1 | subject_id  | A         |           3 |     NULL |
NULL   |      | BTREE      |         |               |
+-------------+------------+-----------------------+--------------+-------------+-----------+-------------+--
--------+--------+------+------------+---------+---------------+
2 rows in set (0.00 sec)

 slave에서 확인

mysql> show index from tb_textbook;
+-------------+------------+-----------------------+--------------+-------------+-----------+-------------+--
--------+--------+------+------------+---------+---------------+
| Table       | Non_unique | Key_name              | Seq_in_index | Column_name | Collation | Cardinality |
Sub_part | Packed | Null | Index_type | Comment | Index_comment |
+-------------+------------+-----------------------+--------------+-------------+-----------+-------------+--
--------+--------+------+------------+---------+---------------+
| tb_textbook |          0 | PRIMARY               |            1 | textbook_id | A         |           3 |     NULL |
NULL   |      | BTREE      |         |               |
| tb_textbook |          1 | textbook_subject_id_idx |          1 | subject_id  | A         |           3 |     NULL |
NULL   |      | BTREE      |         |               |
+-------------+------------+-----------------------+--------------+-------------+-----------+-------------+--
--------+--------+------+------------+---------+---------------+
2 rows in set (0.00 sec)
```

-. now() 와 sysdate() 의 사용에 대한 테스트

now()와 sysdate()의 차이는 MySQL에서의 현재시각조회를 위한 sysdate와 now 의 차이 에서 설명이 되었다. 그중 replication 에서 실제로 어떤 차이가 있는지 확인해 보도록 한다.
먼저 아래와 같은 구조를 가진 임시 table을 생성한다.

```
master mysql> desc tb_timetable;
+---------+-------------+------+-----+---------+-------+
| Field   | Type        | Null | Key | Default | Extra |
+---------+-------------+------+-----+---------+-------+
| nowdate | varchar(50) | NO   |     | NULL    |       |
| sysdate | varchar(50) | NO   |     | NULL    |       |
+---------+-------------+------+-----+---------+-------+
2 rows in set (0.00 sec)
그리고 다음의 쿼리로 now()와 sysdate()를 각각 입력하도록 한다.
INSERT INTO `classmatetest`.`tb_timetable` (`nowdate`,`sysdate`)VALUES (NOW(),
SYSDATE());



master mysql> select * from tb_timetable;
+---------------------+---------------------+
| nowdate             | sysdate             |
+---------------------+---------------------+
| 2013-02-07 14:50:37 | 2013-02-07 14:50:37 |
+---------------------+---------------------+
1 row in set (0.00 sec)

slave mysql> select * from tb_timetable;

+---------------------+---------------------+
| nowdate             | sysdate             |
+---------------------+---------------------+
| 2013-02-07 14:50:37 | 2013-02-07 14:41:31 |
+---------------------+---------------------+
1 row in set (0.00 sec)
```

기본적으로 now와 sysdate는 현재시각을 반환하는 함수로 동일한 기능이다. 하지만 replication에서 now를 사용하게 되면 master와 slave에 동일한 값으로 관리가 되지만, sysdate를 사용하게 되면 호출한 시간 반화하는 관계로 master와 slave에 입력된 값이 다르게 되는 문제점이 있다. 이러한 점을 고려하여 어플리케이션에서 sysdate보다는 now 를 사용하는것이 낫고, 그렇지 못한 상황이라면 mysql 의 --sysdate-is-now 옵션을 설정하여 sysdate()와 now()의 함수를 동일하게 적용하도록 한다.

참조
http://hanaduri.egloos.com/2389708
http://linuxism.tistory.com/846