

LABWORK

COURSE: DISTRIBUTED SYSTEMS

CHAPTER 7: Consistency and Replication

1. Replication and Consistency for MySQL

1.1. Contents

A major problem in every distributed system today is the replication and consistency of data. The data is usually replicated to improve reliability or performance. One of the major problems is to maintain the consistency of these replicas. Informally, this means that when a copy is updated, we must make sure that the other copies are updated as well; otherwise, the replicas will not be the same anymore.

The main questions here are "Why is replication useful? and how does this relate to scalability? What does coherence really mean?". First, we start by focusing on replica management, which takes into account not only the location of replica servers, but also the way content is distributed to these servers. The second problem concerns the consistency of replicas. In most cases, applications require strong consistency. Informally, this means that updates must be propagated more or less immediately between replicas.

In this labwork, we will implement the replication mechanism of MySQL databases. MySQL Replication is a process that allows you to easily keep multiple copies of MySQL data by automatically copying them from a master database to a slave database. This can be useful for a variety of reasons, including facilitating a data backup, a way to analyze it without using the main database, or simply as a way to evolve.

1.2. Requirements

1.2.1. Theory

- Theory on Replication and Consistency

1.2.2. Hardwares

- 2 PCs on Linux

1.2.3. Softwares

- mysql

1.3. PRACTICAL STEPS

In this labwork, you will use 2 machines, Master and Slave, that work on Linux operating system.

Set the network configuration so that these 2 machines can connect to each other.

Install MySQL on both machines:

```
$ sudo apt-get install mysql-server mysql-client
```

① Caution: Never forget the *root* password you have created.

Configuration for Master machine

On the Master machine, you add the following content to the file */etc/mysql/my.cnf*:

```
[mysqld]
bind-address = IP_of_Master
server-id = 1
log_bin = /var/log/mysql/mysql-bin.log
binlog_do_db = petdatabase
```

Restart the MySQL with the command:

```
$sudo service mysql restart
```

Now, log in to the MySQL :

```
$mysql -u root -p
```

(type the root password that you created during the installation step)

Create an account in MySQL for the Slave:

```
mysql> GRANT REPLICATION SLAVE ON *.* TO 'slave_user'@'%'
IDENTIFIED BY 'password';
```

Next:

```
mysql>FLUSH PRIVILEGES;
```

Create a new database named *petdatabase*:

```
mysql>CREATE DATABASE petdatabase;
```

Go to this new database:

```
mysql>USE petdatabase;
```

Create a new table named *pet* in this database:

```
mysql>CREATE TABLE pet (name VARCHAR(20), owner
VARCHAR(20), species VARCHAR(20), sex CHAR(1), birth
DATE, death DATE);
```

Verify this new table:

```
mysql> SHOW TABLES;
mysql> DESCRIBE pet;
```

Exit temporarily with the command:

```
mysql>EXIT;
```

Write a text file to import the content into the *pet* table of the *petdatabase*. Alternatively, you can download it with the link: <https://goo.gl/K6BafW>

Access MySQL and import the contents of this file:

```
$mysql -u root -p
mysql>USE petdatabase;
mysql>LOAD DATA LOCAL INFILE '/path/pet.txt' INTO TABLE pet;
```

ⓘ Caution: You have to replace the right path to the *pet.txt* file.

Check the inserted content:

```
mysql>SELECT * FROM pet;
```

Question 1: What is the output did you see? Now, try to add another entry to the table *pet* in using SQL queries.

Now, you must block the read operation to ensure that nothing is changed until we replicate to the slave machine:

```
mysql> FLUSH TABLES WITH READ LOCK;
```

Show the status of the Master :

```
mysql> SHOW MASTER STATUS;
```

Question 2: What is the name of the log file and the position?

ⓘ Caution: This is the file name and position used by the master server for the log. Write them down because you will have to use them later.

You leave this terminal window (you'll come back after) and make sure that you don't touch it anymore. Open another terminal window for the rest.

Export the database file:

```
$mysqldump -u root -p --opt petdatabase > petdatabase.sql
```

Send this file to the Slave machine:

```
$scp petdatabase.sql username@IP_Slave_machine:.
```

Replace the username of and the IP address of the Slave machine.

Question 3: Have you received this file in Slave machine? What is the path of this received file in the Slave machine?

Configuration for Slave machine

Create the database *petdatabase*:

```
mysql>CREATE DATABASE petdatabase;
mysql>EXIT;
```

Import the received file to the created database:

```
$mysql -u root -p petdatabase < /path/to/petdatabase.sql
```

Edit the content of the file */etc/mysql/my.cnf* :

```
[mysqld]
server-id = 2
relay-log = /var/log/mysql/mysql-relay-bin.log
log_bin = /var/log/mysql/mysql-bin.log
binlog_do_db = petdatabase
```

Restart MySQL:

```
$sudo service mysql restart
```

Realize the replication:

```
mysql>CHANGE MASTER TO MASTER_HOST='12.34.56.789',
MASTER_USER='slave_user', MASTER_PASSWORD='password',
MASTER_LOG_FILE='mysql-bin.000001', MASTER_LOG_POS= 107;
```

(change all the appropriate values in bold format)

This command above accomplishes several things at the same time:

- It designates the current server as slave of our master server.
- It provides the server with the correct login information

- Finally, it tells the slave server where to begin replication. The main log file and the log position come from the numbers you obtained on Master's machine.

Now, activate the Slave:

```
mysql>START SLAVE;
```

Show the status of the Slave:

```
mysql>SHOW SLAVE STATUS\G
```

Question 4: What is the status information you received? How do you know the configuration is OK?

Test the replication

Now, you can return to the shell window on the Master machine that you leaved it blocked. Type:

```
mysql>UNLOCK TABLES;
```

Try to insert new data to database petdatabase:

```
mysql> INSERT INTO pet VALUES  
( 'Puffball', 'Diane', 'hamster', 'f', '1999-03-30', NULL );
```

Question 5: In the Slave machine, verify if the new inserted data has been replicated from Master to Slave. Which command did you use?