3.20 实战案例——MySQL 服务器的运维与优化

3.20.1 案例目标

（1）了解 MySQL 的基础命令。

（2）使用 MySQL 基础运维命令。

（3）了解 MySQL 数据库优化。

3.20.2 案例分析

1. 规划节点

MySQL 服务运维与排错的节点规划，见表 3-20-1。

表 3-20-1 规划节点

IP 地址		节点
172.30.15.10	**mysql**	**mysql**

2. 基础准备

使用 VMWare Workstation 软件安装 CentOS 7.2 操作系统，镜像使用提供的 CentOS-7-x86_64-DVD-1511.iso，最小化安装 CentOS 7.2 系统，YUM 源使用提供的本地 gpmall-repo 包，安装基础环境。

3.20.3 案例实施

1. MySQL 运维

（1）安装数据库

配置本地 YUM 安装源，将提供的 gpmall-repo 文件上传至/opt 目录，创建 local.repo 文件，示例代码如下：（若使用的是 VMware 安装的 CentOS 7.2 系统，自带的 CentOS.repo 文件不要移除。若使用的是 OpenStack 中的 centos7.2qcow2 镜像需要将自带的 CentOS.repo 文件移除。）

```
[root@localhost ~]# vi /etc/yum.repos.d/yum.repo
[mariadb]
name=mariadb
baseurl=file:///opt/mariadb_yum/
```

```
gpgcheck=0
enabled=1
[root@localhost ~]# yum install -y mariadb mariadb-server
[root@localhost ~]# systemctl start mariadb
[root@localhost ~]# mysql_secure_installation
NOTE: RUNNING ALL PARTS OF THIS SCRIPT IS RECOMMENDED FOR ALL MariaDB
      SERVERS IN PRODUCTION USE!  PLEASE READ EACH STEP CAREFULLY!

In order to log into MariaDB to secure it, we'll need the current
password for the root user.  If you've just installed MariaDB, and
you haven't set the root password yet, the password will be blank,
so you should just press enter here.

Enter current password for root (enter for none):
OK, successfully used password, moving on...

Setting the root password ensures that nobody can log into the MariaDB
root user without the proper authorisation.

Set root password? [Y/n] y
New password:
Re-enter new password:
Password updated successfully!
Reloading privilege tables..
 ... Success!

By default, a MariaDB installation has an anonymous user, allowing anyone
to log into MariaDB without having to have a user account created for
```

```
them.  This is intended only for testing, and to make the installation
go a bit smoother.  You should remove them before moving into a
production environment.

Remove anonymous users? [Y/n] y
 ... Success!

Normally, root should only be allowed to connect from 'localhost'.  This
ensures that someone cannot guess at the root password from the network.

Disallow root login remotely? [Y/n] n
 ... skipping.

By default, MariaDB comes with a database named 'test' that anyone can
access.  This is also intended only for testing, and should be removed
before moving into a production environment.

Remove test database and access to it? [Y/n] y
 - Dropping test database...
 ... Success!
 - Removing privileges on test database...
 ... Success!

Reloading the privilege tables will ensure that all changes made so far
will take effect immediately.

Reload privilege tables now? [Y/n] y
 ... Success!
```

```
Cleaning up...


All done!   If you've completed all of the above steps, your MariaDB
installation should now be secure.


Thanks for using MariaDB!
```

（2）创建数据库

创建一个名称为“test”数据库。命令如下所示：

```
[root@mysql ~]# mysqladmin -uroot -p000000 create test
```

在“test”数据库中创建一个名为“tables”数据表。命令如下所示：

```
MariaDB [(none)]> use test;
Database changed
MariaDB [test]> CREATE TABLE IF NOT EXISTS `tables`(
`tables_id` INT UNSIGNED AUTO_INCREMENT,
                `tables_title` VARCHAR(100) NOT NULL,
                `tables_author` VARCHAR(40) NOT NULL,
                `tables_date` DATE,
                PRIMARY KEY ( `tables_id` )
)ENGINE=InnoDB DEFAULT CHARSET=utf8;



+----------------+
| Tables_in_test |
+----------------+
| tables         |
+----------------+
1 row in set (0.001 sec)
```

（3）数据库备份

导出整个数据库，命令如下所示。

```
[root@mysql ~]# mysqldump -uroot -p000000 test > test.sql
[root@mysql ~]# ls
test.sql
```

导出一个表，命令如下所示：

```
[root@mysql ~]# mysqldump -uroot -p000000 test tables > test_tables.sql
[root@mysql ~]# ls
test.sql    test_tables.sql
```

删除test数据库进行导入测试，用mysqldump备份的文件是一个可以直接导入的SQL脚本。有两种方法可以将数据导入，一种用msql命令，把数据库文件恢复到指定的数据库，命令如下所示：

```
[root@mysql ~]# mysqladmin -uroot -p000000 drop test
Dropping the database is potentially a very bad thing to do.
Any data stored in the database will be destroyed.

Do you really want to drop the 'test' database [y/N] y
Database "test" dropped
[root@mysql ~]# mysql -uroot -p000000
Welcome to the MariaDB monitor.  Commands end with ; or \g.
Your MariaDB connection id is 26
Server version: 10.3.18-MariaDB-log MariaDB Server

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)]> create database test;
Query OK, 1 row affected (0.000 sec)

MariaDB [(none)]> quit
```

```
Bye
[root@mysql ~]# mysql -uroot -p000000 test < test.sql
```

第二种，可以使用 source 语句方法导入数据库，把数据库文件恢复到指定的数据库，命令如下所示：

```
[root@mysql ~]# mysqladmin -uroot -p000000 drop test
Dropping the database is potentially a very bad thing to do.
Any data stored in the database will be destroyed.

Do you really want to drop the 'test' database [y/N] y
Database "test" dropped
[root@mysql ~]# mysql -uroot -p000000
Welcome to the MariaDB monitor.  Commands end with ; or \g.
Your MariaDB connection id is 30
Server version: 10.3.18-MariaDB-log MariaDB Server
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)]> create database test;
Query OK, 1 row affected (0.027 sec)

MariaDB [(none)]> use test
Database changed
MariaDB [test]> source /root/test.sql;
```

（4）添加用户并授权

授权 root 用户可以在任何节点访问 test 数据库下所有表，“%”代表所有节点机器，命令如下所示：

```
MariaDB [(none)]> GRANT ALL PRIVILEGES ON test.* TO 'root'@'%' IDENTIFIED BY '000000' ;
```

```
Query OK, 0 rows affected (0.001 sec)
MariaDB [(none)]> GRANT ALL PRIVILEGES ON test.* TO 'root'@'localhost' IDENTIFIED BY '000000' ;
Query OK, 0 rows affected (0.001 sec)
```

添加 root 用户对 test 数据库授增、删、改、查的权限，命令如下所示：

```
MariaDB [(none)]> GRANT SELECT,INSERT,DELETE,UPDATE ON test.* TO 'root'@'%' IDENTIFIED BY '000000' ;
Query OK, 0 rows affected (0.001 sec)
```

2. MySQL 数据库优化

修改数据库配置文件，添加参数，命令如下所示：

```
# vi /etc/my.cnf
```

在文件中添加以下命令，优化数据库：

```
[mysqld]
thread_concurrency = 64            #CPU 核数  * 2
max_connections=1500               #最大连接（用户）数。每个连接 MySQL 的用户均算作一个连接
max_connect_errors=30              #最大失败连接限制
bulk_insert_buffer_size = 32M      #批量插入数据缓存大小
query_cache_type=1                 #查询缓存   (0 = off、1 = on、2 = demand)
query_cache_size = 64M             #指定 mysql 查询缓冲区大小
max_allowed_packet = 128M          #通信缓冲大小
read_buffer_size = 8M              #顺序读取数据缓冲区使用内存
read_rnd_buffer_size = 32M         #随机读取数据缓冲区使用内存
```

参数优化解析，见表 3-20-2。

表 3-20-2 参数优化命令解析

命令	解析
thread_concurrency	并发线程数，建议为 CPU 核心数乘以 2
max_connections	最大连接（用户）数。每个连接 MySQL 的用户均算作一个连接
max_connect_errors	最大失败连接限制
bulk_insert_buffer_size	批量插入数据缓存大小，可以有效提高写入效率，默认为 8 MB

query_cache_type	控制着查询缓存功能的开启的关闭。0 时表示关闭，1 时表示打开，2 表示只要 select 中明确指定 SQL_CACHE 才缓存
query_cache_size	指定 MySQL 查询缓冲区的大小，用来缓冲 select 的结果，并在下一次同样查询的时候不再执行查询而直接返回结果，根据 Qcache_lowmem_prunes 的大小，来查看当前的负载是否足够高，在数据库写入量或是更新量也比较大的系统，该参数不适合分配过大。而且在高并发，写入量大的系统，建议把该功能禁掉。属重点优化参数（主库增删改-MyISAM）
max_allowed_packet	设定在网络传输中一次可以传输消息的最大值，系统默认为 1 MB，最大可 1 GB
read_buffer_size	来做 MYISAM 表全表扫描的缓冲大小，对表进行顺序扫描的请求将分配一个读入缓冲区，MySQL 会为它分配一段内存缓冲区。read_buffer_size 变量控制这一缓冲区的大小。如果对表的顺序扫描请求非常频繁，并且用户认为频繁扫描进行得太慢，可以通过增加该变量值以及内存缓冲区大小提高其性能
read_rnd_buffer_size	随机读（查询操作）缓冲区大小。当按任意顺序读取行时(例如，按照排序顺序)，将分配一个随机读缓存区。进行排序查询时，MySQL 会首先扫描一遍该缓冲，以避免磁盘搜索，提高查询速度，如果需要排序大量数据，可适当调高该值。但 MySQL 会为每个客户连接发放该缓冲空间，所以应尽量适当设置该值，以避免内存开销过大