

MySQL学习笔记（Day006：存储引擎二/多实例安装）

MySQL 学习

MySQL学习笔记（Day006：存储引擎二/多实例安装）

- 一. MyISAM存储引擎(下)
 - 1. MyISAM还在使用的原因
 - 2. MyISAM文件组成
 - 3. myisamchk
- 二. Memory存储引擎
 - 1. Memory介绍
 - 2. Memory特性
 - 3. Memory的物理特性
- 三. CSV存储引擎
 - 1. CSV介绍
 - 2. CSV文件组成
 - 2. CSV特性
- 四. Federated存储引擎
 - 1. Federated介绍
 - 2. Federated 语法
- 五. 多实例安装
 - 1. 多实例介绍
 - 2. 安装要求
 - 3. 安装操作
- 六. Federated测试

一. MyISAM存储引擎(下)

1. MyISAM还在使用的原因

- 历史原因，需要逐步替换
- 部分如User，DB等系统表(MyISAM引擎)，可以直接拷贝，比较方便
- 性能好，或者存储小 **不是** MyISAM的优点，也不是存在的原因

2. MyISAM文件组成

- **frm** 表结构文件
- **MYI** 索引文件
- **MYD** 数据文件
 - 数据文件是堆表数据结构，堆是无序数据的集合
 - **MYI** 中的叶子节点，指向 **MYD** 中的数据页
 - 当数据移动到页外时，需要修改对应指针

3. myisamchk

myisamchk 通过扫描MYD文件来重建MYI文件；如果MYD文件中某条记录有问题，将跳过该记录

二. Memory存储引擎

1. Memory介绍

- 全内存存储的引擎
- 数据库重启后数据丢失
- 支持哈希索引
- 不支持事物

2. Memory特性

- 千万不要用**Memory**存储引擎去做缓存(**Cache**)，**性能上不及Redis和Memcached**
- Memory 不能禁用，当涉及内部排序操作的临时表时，使用该存储引擎
 - **max_heap_table_size** 决定使用内存的大小，默认时 **1G**
 - 无论该表使用的什么引擎，只要使用到临时表，或者指定Memory，都受参数影响
 - 当上面设置的内存放不下数据时，(>=5.6)转为MyISAM(>=5.7)转为InnoDB
 - 注意磁盘上临时路径空间的大小(**tmpdir**)
 - 内存使用为会话(SESSION)级别，当心内核OOM
- 支持哈希索引，且仅支持等值查询

```
mysql> show global status like "%tmp tables";
+-----+
| Variable_name | Value |
+-----+
+-----+
| Created_tmp_disk_tables | 0 | -- 内存放不下，转成磁盘存储的数量,如果过大，考虑增大内存参数
| Created_tmp_tables | 4 | -- 创建临时表的数量
+-----+
2 rows in set (0.00 sec)

mysql> show variables like 'tmpdir';
+-----+
| Variable_name | Value |
+-----+
+-----+
| tmpdir | /tmp | -- memory转成磁盘存储的路径
+-----+
1 row in set (0.00 sec)

mysql> show create table User\G
***** 1. row *****
Table: User
Create Table: CREATE TABLE 'User' (
  'id' int(11) NOT NULL,
  'name' varchar(128) DEFAULT NULL,
  PRIMARY KEY ('id'),
  KEY 'name' ('name') USING HASH -- 对这个字段使用USING HASH,创建hash索引
) ENGINE=MEMORY DEFAULT CHARSET=latin1
1 row in set (0.00 sec)
```

3. Memory的物理特性

- 内存不会一次性分配最大空间，而是随着使用逐步增到到最大值
- 通过链表管理空闲空间
- 使用固定长度存储数据
- 不支持BLOB和TEXT类型
- 可以创建自增主键

三. CSV存储引擎

1. CSV介绍

- CSV - Comma-Separated Values，使用逗号分隔
- 不支持特殊字符
- CSV是一种标准文件格式
- 文件以纯文本形式存储表格数据
- 使用广泛

2. CSV文件组成

- **frm** 表结构
- **CSV** 数据文件
- **CSN** 元数据信息

2. CSV特性

- MySQL CSV存储引擎运行时，即创建CSV 文件
- 通过MySQL标准接口来查看和修改CSV文件
- 无需将CSV文件导入到数据库，只需创建相同字段的表结构，拷贝CSV文件即可
- CSV存储引擎表每个字段 **必须是NOT NULL** 属性

四.Federated存储引擎

1. Federated介绍

- 允许本地访问远程MySQL数据库中表的数据
- 本地不存储任何数据文件
- 类似Oracle中的DBLink
- Federated存储引擎默认不开启,需要在 **my.cnf** 的 **[mysqld]** 标签下添加 **federated**
- MySQL的Federated不支持异构数据库访问，MariaDB中的 **Federatedx** 支持

2. Federated 语法

```
scheme://user_name[:password]@host_name[:port_num]/db_name/tbl_name

CONNECTION='mysql://username:password@hostname:port/database/tablename'

--
-- 例子
--
CREATE TABLE `T1` (
  `A` VARCHAR(100),
  UNIQUE KEY (`A` (30))
) ENGINE=FEDERATED
CONNECTION='MYSQL://@avid:123@127.0.0.1:3306/TEST/T1';
```

五. 多实例安装

1. 多实例介绍

- 一台服务器上安装多个MySQL数据库实例
- 可以充分利用服务器的硬件资源
- 通过mysqld_multi进行管理

2. 安装要求

- MySQL实例1 - `mysql1`
 - port = 3306
 - datadir = /data1
 - socket = /tmp/mysql.sock1
- MySQL实例2 - `mysql2`
 - port = 3307
 - datadir = /data2
 - socket = /tmp/mysql.sock2
- MySQL实例3 - `mysql3`
 - port = 3308
 - datadir = /data3
 - socket = /tmp/mysql.sock3
- MySQL实例4 - `mysql4`
 - port = 3309
 - datadir = /data4
 - socket = /tmp/mysql.sock4

该三个参数必须定制，且必须不同（port / datadir / socket）
`server-id` 和多数数据库实例没有关系，和数据库复制有关系。

3. 安装操作

```
#
# 多实例配置文件，可以mysqld_multi --example 查看例子
#
[root@HyServer ~]# cat /etc/my.cnf
#(client)
# 这个标签如果配置了用户和密码，
# 并且[mysqld_multi]下没有配置用户名密码，
# 则mysqld_multi stop时，会使用这个密码
# 如果没有精确的匹配，则匹配[client]标签

#user = root
#password = 123
#-----
[mysqld_multi]
mysqld = /usr/local/mysql/bin/mysqld_safe
mysqladmin = /usr/local/mysql/bin/mysqladmin
user = multi_admin
pass = 123 # 官方文档中写的password，但是存在bug，需要改成pass(v5.7.9)
# 写成password，start时正常，stop时，报如下错误
# Access denied for user 'multi_admin'@'localhost' (using password: YES)
log = /var/log/mysqld_multi.log

[mysqld1] # mysqld后面的数字为GNR，是该实例的标识
# mysqld_multi start 1, mysqld_multi start 2-4
server-id = 11
socket = /tmp/mysql.sock1
port = 3306
bind_address = 0.0.0.0
datadir = /data1
user = mysql
performance_schema = off
innodb_buffer_pool_size = 32M
skip_name_resolve = 1
log_error = error.log
pid-file = /data1/mysql.pid1

[mysqld2]
server-id = 12
socket = /tmp/mysql.sock2
port = 3307
bind_address = 0.0.0.0
datadir = /data2
user = mysql
performance_schema = off
innodb_buffer_pool_size = 32M
skip_name_resolve = 1
log_error = error.log
pid-file = /data2/mysql.pid2

[mysqld3]
server-id = 13
socket = /tmp/mysql.sock3
port = 3308
bind_address = 0.0.0.0
datadir = /data3
user = mysql
performance_schema = off
innodb_buffer_pool_size = 32M
skip_name_resolve = 1
log_error = error.log
pid-file = /data3/mysql.pid3

[mysqld4]
server-id = 14
socket = /tmp/mysql.sock4
port = 3309
bind_address = 0.0.0.0
datadir = /data4
user = mysql
performance_schema = off
innodb_buffer_pool_size = 32M
skip_name_resolve = 1
log_error = error.log
pid-file = /data4/mysql.pid4

#
# 准备好数据目录，并初始化安装
#
[root@HyServer ~]# mkdir /data1
[root@HyServer ~]# mkdir /data2
[root@HyServer ~]# mkdir /data3
[root@HyServer ~]# mkdir /data4
[root@HyServer ~]# chown mysql.mysql /data{1..4}
[root@HyServer ~]# mysqld --initialize --user=mysql --datadir=/data1
#
# 一些日志输出，并提示临时密码，下同
#
[root@HyServer ~]# mysqld --initialize --user=mysql --datadir=/data2
[root@HyServer ~]# mysqld --initialize --user=mysql --datadir=/data3
[root@HyServer ~]# mysqld --initialize --user=mysql --datadir=/data4
# 安装后，需要检查error.log 确保没有错误出现
[root@HyServer ~]# cp /usr/local/mysql/support-files/mysqld_multi.server /etc/init.d/mysqld_multid
# 将见启动脚本，方便自启
[root@HyServer ~]# chkconfig mysqld_multid on

[root@HyServer ~]# mysqld_multi start
[root@HyServer ~]# mysqld_multi report
Reporting MySQL servers
MySQL server from group: mysqld1 is running
MySQL server from group: mysqld2 is running
MySQL server from group: mysqld3 is running
MySQL server from group: mysqld4 is running
[root@HyServer ~]# netstat -tunlp | grep mysql
[root@HyServer ~]# netstat -tunlp | grep mysql

tcp      0      0 :::3307                :::*                    LISTEN    6221/mysql
tcp      0      0 :::3308                :::*                    LISTEN    6232/mysql
tcp      0      0 :::3309                :::*                    LISTEN    6238/mysql
tcp      0      0 :::3306                :::*                    LISTEN    6201/mysql

[root@HyServer ~]# mysql -u root -S /tmp/mysql.sock1 -p -P3306
#
# 使用-S /tmp/mysql.sock1 进行登录，并输入临时密码后，修改密码，下同
#
[root@HyServer ~]# mysql -u root -S /tmp/mysql.sock2 -p -P3307
[root@HyServer ~]# mysql -u root -S /tmp/mysql.sock3 -p -P3308
[root@HyServer ~]# mysql -u root -S /tmp/mysql.sock4 -p -P3309
```



```
--
-- mysql1
--
mysql> show variables like "port";
+-----+
| Variable_name | Value |
+-----+
| port          | 3306  |
+-----+
1 row in set (0.00 sec)

mysql> show variables like "socket";
+-----+
| Variable_name | Value |
+-----+
| socket        | /tmp/mysql.sock1 |
+-----+
1 row in set (0.01 sec)

mysql> show variables like "datadir";
+-----+
| Variable_name | Value |
+-----+
| datadir       | /data1/ |
+-----+
1 row in set (0.00 sec)

--
-- 这样才能进行关闭数据库的操作
-- 和(mysql多实例)中的user、pass(注意在5.7.9中不是password)对应起来 (类似[client]标签)
-- 一会测试federated连接, 需要增加federated参数, 并重启mysql2
--

mysql> create user 'multi_admin'@'localhost' identified by '123';
Query OK, 0 rows affected (0.00 sec)
mysql> grant shutdown on *.* to 'multi_admin'@'localhost';

--
-- mysql2, mysql3, mysql4 类似, 可以看到与my.cnf中对应的port和socket
--
```

六. Federated测试

```
--
-- mysql1 准备数据
--
mysql> show variables like "port";
+-----+
| Variable_name | Value |
+-----+
| port          | 3306  | -- mysql1 实例端口
+-----+
1 row in set (0.00 sec)

mysql> create database burn;
Query OK, 1 row affected (0.00 sec)

mysql> use burn;
Database changed

mysql> create table book (
    -> id int not null auto_increment,
    -> name varchar(128) not null,
    -> primary key(id)
    -> );
Query OK, 0 rows affected (0.20 sec)

mysql> insert into book values(1, "book1");
Query OK, 1 row affected (0.02 sec)

mysql> select * from book;
+-----+
| id | name |
+-----+
| 1  | book1 |
+-----+
1 row in set (0.00 sec)

mysql> create user 'burn'@'127.0.0.1' identified by '123';
Query OK, 0 rows affected (0.00 sec)

mysql> grant select on burn.* to 'burn'@'127.0.0.1';
Query OK, 0 rows affected (0.00 sec)

mysql> show grants for 'burn'@'127.0.0.1';
+-----+
| Grants for burn@127.0.0.1 |
+-----+
| GRANT USAGE ON *.* TO 'burn'@'127.0.0.1' |
| GRANT SELECT ON 'burn'.* TO 'burn'@'127.0.0.1' |
+-----+
2 rows in set (0.00 sec)

--
-- mysql2 测试Federated
--
mysql> show variables like "port";
+-----+
| Variable_name | Value |
+-----+
| port          | 3307  | -- mysql2 实例端口
+-----+
1 row in set (0.01 sec)

mysql> show engines;
+-----+
| Engine | Support | Comment | Transactions | XA | Savepoints |
+-----+
| MyISAM | YES | MyISAM storage engine | NO | NO | NO |
| CSV | YES | CSV storage engine | NO | NO | NO |
| PERFORMANCE_SCHEMA | YES | Performance Schema | NO | NO | NO |
| BLACKHOLE | YES | /dev/null storage engine (anything you write to it disappears) | NO | NO | NO |
| MRG_MYISAM | YES | Collection of identical MyISAM tables | NO | NO | NO |
| InnoDB | DEFAULT | Supports transactions, row-level locking, and foreign keys | YES | YES | YES |
| ARCHIVE | YES | Archive storage engine | NO | NO | NO |
| MEMORY | YES | Hash based, stored in memory, useful for temporary tables | NO | NO | NO |
| FEDERATED | NO | Federated MySQL storage engine | NULL | NULL | NULL |
+-----+

--
-- federated 引擎没有打开
--
9 rows in set (0.00 sec)

#
# 在[mysqld2]标签下面增加federated
#

[root@MyServer ~]> cat /etc/my.cnf
# ... 省略 ...
[mysqld2]
federated # 新增的配置项, 表示打开federated引擎
# ... 省略 ...
[root@MyServer ~]> mysqld_multi stop 2
[root@MyServer ~]> mysqld_multi start 2 # 重启配置

mysql> show variables like "port";
+-----+
| Variable_name | Value |
+-----+
| port          | 3307  | -- mysql2 实例端口
+-----+
1 row in set (0.01 sec)

mysql> show engines;
+-----+
| Engine | Support | Comment | Transactions | XA | Savepoints |
+-----+
| MyISAM | YES | MyISAM storage engine | NO | NO | NO |
| CSV | YES | CSV storage engine | NO | NO | NO |
| PERFORMANCE_SCHEMA | YES | Performance Schema | NO | NO | NO |
| BLACKHOLE | YES | /dev/null storage engine (anything you write to it disappears) | NO | NO | NO |
| MRG_MYISAM | YES | Collection of identical MyISAM tables | NO | NO | NO |
| InnoDB | DEFAULT | Supports transactions, row-level locking, and foreign keys | YES | YES | YES |
| ARCHIVE | YES | Archive storage engine | NO | NO | NO |
| MEMORY | YES | Hash based, stored in memory, useful for temporary tables | NO | NO | NO |
| FEDERATED | YES | Federated MySQL storage engine | NO | NO | NO |
+-----+
9 rows in set (0.00 sec)

--
-- 显示 federated 已经启用
--
mysql> create database federated_test;
Query OK, 1 row affected (0.00 sec)

mysql> use federated_test;
Database changed

mysql> create table federated_table_1 (
    -> id int not null auto_increment,
    -> name varchar(128) not null,
    -> primary key(id)
    -> ) engine=federated
    -> connection='mysql://burn:123@127.0.0.1:3306/burn/book';
Query OK, 0 rows affected (0.04 sec)

mysql> select * from federated_table_1;
+-----+
| id | name |
+-----+
| 1  | book1 | -- 和 mysql1d1 上的内容一致。
+-----+
1 row in set (0.00 sec)

--
-- 由于只有select权限, 无法对该表进行insert操作
--
mysql> insert into federated_table_1 values(2, "book2");
ERROR 1296 (HY000): Got error 10000 'Error on remote system: 1142: INSERT command denied to user 'burn'@'127.0.0.1' for table 'book1' from FEDERATED
```