

# 实现一个 MySQL 配置对比脚本需要考虑哪些细节？

原创 李彬 爱可生开源社区 2025年04月22日 17:00 上海

作者：李彬，爱可生 DBA 团队成员，负责项目日常问题处理及公司平台问题排查。爱好有亿点点多，吉他、旅行、打游戏

爱可生开源社区出品，原创内容未经授权不得随意使用，转载请联系小编并注明来源。

本文约 1500 字，预计阅读需要 3 分钟。



## 引言

想必在日常运维 MySQL 的过程中，无论是主从搭建、版本升级、数据迁移，还是定期巡检，我们总会遇到需要对比 MySQL 配置的场景。这种看似简单的需求，实际操作中隐藏着很多细节和挑战。

虽然已有 `pt-config-diff`<sup>[1]</sup> 类似的工具可以使用，但在一些定制化运维场景中，不可避免地需要自己去实现相关代码。

本文便以实现 MySQL 配置对比的功能为例，与大家探讨脚本中需要考虑的关键点和容易被忽略的细节。让我们开始下面的内容。

**模拟场景：**针对同一实例，如何对比变量的运行值、配置文件值？

## 1. 运行值的获取

既然要对比变量的运行值与配置文件中的配置值（后文均以运行值和配置值简述），我们便从如何获取对应的值说起。

首先，获取变量的运行值。

```
[root@localhost ~]$ /opt/sandboxes/mysql/5.7.31/bin/mysql --login-path=root --socket=/tmp  
REPEATABLE-READ
```

当然，你也可以直接查询 `performance_schema.global_variables` 表。

```
[root@localhost ~]$ /opt/sandboxes/mysql/5.7.31/bin/mysql --login-path=root --socket=/tmp  
REPEATABLE-READ
```

可以看到，运行值的获取较为简单。当然，在通过脚本执行时，不可避免地需要使用明文密码，这里有两个建议：

- 添加 `--login-path=root` 参数，相比明文密码安全一些。
- 创建专用用户，使用最小权限。

## 2. 配置文件的解析

在说明配置值如何获取前，我们考虑下除了运行值，还有哪些可能影响变量的文件？

### my.cnf

推荐的运维场景下，一般使用 `--defaults-file` 参数指定配置文件，这样做有几点好处：

- 写脚本时方便我们使用进程信息来过滤 `--defaults-file` 获取配置文件的位置。
- 可以避免受到系统默认的 `/etc/my.cnf` 等其他配置文件的干扰。

### mysqld-auto.cnf

除了默认的配置文件，在 MySQL 8.0 中，还需考虑被持久化的变量配置文件，即 `mysqld-auto.cnf`。

有时我们会在启动 `mysqld` 进程时手工指定某些参数配置，此时也会影响对应的变量值。这种启动方式一般仅用于临时维护使用，本文暂不考虑。

针对 `my.cnf` 文件中配置值的获取，这里我提供一些需要考虑的场景供大家参考：

1. 如 `port`、`socket` 等变量配置，在 `[mysql]`、`[mysqld]` 等标签中均有配置，在获取时需要先过滤出配置文件中关于 `[mysqld]` 的部分。

2. 同一变量，在配置文件中可能配置多次，故解析后仅需拿到最后一次配置的值。
3. 格式问题。在生产环境中的配置文件中，可能出现各种格式，故需要先对配置文件进行整体处理，再过滤到对应的 `key` 和 `value`。常见的有：
  - 大小写问题。需将配置文件的内容统一转换为大写或小写后，再进行匹配过滤。
  - 注释问题。需过滤掉注释行，同时还要注意 `port=3306 # 端口` 格式的注释，在获取值时需要使用 `awk` 或 `sed` 进行处理。
  - 空格问题。可优先处理配置文件中各个位置的空格，最终呈现出 `key=value` 这种紧凑的格式后再进行匹配过滤。
  - “\_”与“-”。目前 MySQL 兼容两种写法，如 `binlog_format` 或 `binlog-format`，在匹配时可以使用正则表达式的 `.` 进行模糊匹配。
  - 考虑 `loose` 开头的变量。

至此，我们已经获取了 `my.cnf` 配置文件中的变量名及其对应的值。接下来我们一起来看看 `mysqld-auto.cnf`，以下是一个简化的 `mysqld-auto.cnf` 文件内容示例：

```
[root@localhost data]$ cat mysqld-auto.cnf | jq
{
  "Version": 2,
  "mysql_static_variables": {
    "innodb_buffer_pool_size": {
      "Value": "1073741824",
      "Metadata": {
        "Host": "localhost",
        "User": "msandbox",
        "Timestamp": 1734599734016339
      }
    }
  },
  "mysql_dynamic_parse_early_variables": {
    "max_connections": {
      "Value": "5000",
      "Metadata": {
        "Host": "localhost",
        "User": "msandbox",
        "Timestamp": 1734599700972489
      }
    }
  }
}
```

这里提供两种思路获取对应的 `key` 和 `value`。

## 使用 `awk` 进行正则匹配

```
[root@localhost data]$ cat mysqld-auto.cnf | jq | awk '
function clean_str(s) {
    gsub(/^ *"|'|:|,$/, "", s)
    return s
}
/^ *"[^"]+": *$/ {
    k = clean_str($1)
}
/^ *"Value": *"[^"]+"/ {
    sub(/^ *"Value": *"/, "", $0)
    print k ":" clean_str($0)
}
'
```

# 输出示例

```
innodb_buffer_pool_size:1073741824
max_connections:5000
```

## 使用 `jq` 命令

```
[root@localhost data]$ jq -r 'to_entries[] | select(.value | type == "object") | .value |
```

# 输出示例

```
innodb_buffer_pool_size:1073741824
max_connections:5000
```

## 3. 变量值的对比

在我们已经获取了运行值、不同配置文件的有效配置值之后，如何来进行对比呢？如果你想到的只是使用 `==` 进行判断，或许结果会有很大一部分存在误判，你还需要考虑以下几点：

1. 首先是 1 和 ON、0 和 OFF 的兼容性写法。因为 1 或者 0 可能存在其他的含义，比如表示数量或者大小，所以我们可以将 ON 和 OFF 全部转化为 1 和 0 之后再进行对比。
2. 数值 Buffer 类的单位换算。如配置文件中可以配置为 1024、1024K、1024KB、1024M、1024MB 等。
3. `slave` 与 `master`、`source` 与 `replica`，考虑两种情况共存、仅存在 `salve` 或 `master`、后续版本仅存在 `source` 和 `replica` 等情况。
4. 对比前对 key 和 value 的大小写进行统一。

## 4. 总结

在 MySQL 配置对比的实现中，你需要综合考虑多种因素，不断地优化完善代码，才能实现准确可靠的配置对比功能。

本文旨在提供一些较为常见的实现思路，如有其他场景，欢迎大家一起交流。

### 参考资料

[1] pt-config-diff: <https://docs.percona.com/percona-toolkit/pt-config-diff.html>

本文关键字：#MySQL# #配置文件# #pt-config-diff# #my.cnf#

---

**MySQL 如何实现安全连接？**

**DML 误操作？MySQL 闪回工具大盘点**

**如何准确获取 MySQL 主从延迟时间？**

**事务持续执行之谜：怎样找出对行记录上锁的 SQL？**

- 🌟 Github : <https://github.com/actiontech/sqlc>
- 📖 文档 : <https://actiontech.github.io/sqlc-docs/>
- 🌐 官网 : <https://opensource.actionsky.com/sqlc/>
- 👥 微信群 : 请添加小助手加入 ActionOpenSource
- 🔗 商业支持 : <https://www.actionsky.com/sqlc>

MySQL 231    自动化脚本 1    pt-config-diff 1    配置文件 2    my.cnf 1

MySQL · 目录

上一篇

事务持续执行之谜：怎样找出对行记录上锁的 SQL？

下一篇

如何巧妙解决 Too many connections 报错？

修改于2025年05月22日