



首页 / MySQL8.0参数配置不生效问题排查诊断



MySQL8.0参数配置不生效问题排查诊断

原创 闫建(Rock Yan) 云和恩墨技术服务团队 2024-04-01

2169



适用范围

MySQL 8.0

问题概述

客户MySQL8.0环境中，主库参数文件配置innodb_buffer_pool_size为48G，经过后期运维重启后，巡检时发现MySQL内存只分配了8G，和参数/etc/my.cnf配置中显示指定的48G相差甚远，故怀疑运维人员后期在线设置过该参数（MySQL 5.7版本以后 innodb_buffer_pool_size可以在线动态调整），经过再三确认，客户肯定并没有任何运维人员进行过参数调整。

问题分析

1. 确认参数文件中内存设置是否正确？

```
[root@iZf8zcal7db1wnlk0w6a8lZ ~]#  
[root@iZf8zcal7db1wnlk0w6a8lZ ~]# cat /etc/my.cnf | grep -i innodb_buffer_pool_size  
innodb_buffer_pool_size = 48G  
[root@iZf8zcal7db1wnlk0w6a8lZ ~]#
```

2. 确认引用的参数文件是否还有其他选择，是否还存在其他的参数文件？



Rock Yan

LV.3

关注

45
文章

128
粉丝

109K+
浏览量

- 获得了 223 次点赞
- 内容获得 61 次评论
- 获得了 371 次收藏

TA的专栏



关于MySQL的那些事儿
收录 39 篇内容

热门文章

- 升级OpenSSL：CVE-2016-2183漏洞处理解决方案
2024-04-15 10943浏览
- MySQL参数优化系列之- join_buffer_size
2023-05-16 8522浏览
- MySQL8.0的 UNDO 表空间管理
2020-03-08 5348浏览
- 采用keepalived（VIP）作为MySQL主从高可用架构时的一些建议
2023-04-30 5292浏览
- MySQL DBA 日常运维常用命令总结
2024-04-30 4863浏览

在线实训环境入口



MySQL在线实训环境

查看详情 »

最新文章

```
[root@iZf8zcal7db1wnlk0w6a8lZ ~]# find / -name my.cnf
/etc/my.cnf
[root@iZf8zcal7db1wnlk0w6a8lZ ~]#
[root@iZf8zcal7db1wnlk0w6a8lZ ~]# find / -name my.*
/usr/lib/python3.6/site-packages/babel/locale-data/my.dat
/etc/my.cnf
```

说明：通过OS中查找参数文件，并未发现有其他参数文件引用。

3. 在线查询数据库中内存参数大小进行确认。

```
[root@iZf8zcal7db1wnlk0w6a8lZ ~]#
[root@iZf8zcal7db1wnlk0w6a8lZ ~]# /usr/local/mysql/bin/mysql -uroot -p
Enter password:
Welcome to the MySQL monitor.  Commands end with ; or \g.
Your MySQL connection id is 193
Server version: 8.0.36 MySQL Community Server - GPL

Copyright (c) 2000, 2024, Oracle and/or its affiliates.

Oracle is a registered trademark of Oracle Corporation and/or its
affiliates. Other names may be trademarks of their respective
owners.

Type 'help;' or '\h' for help. Type '\c' to clear the current input statement.

root@localhost:(none) 02:47:39 >show global variables like '%innodb_buffer_pool
+-----+-----+
| Variable_name      | Value      |
+-----+-----+
| innodb_buffer_pool_size | 8589934592 |
+-----+-----+
1 row in set (0.01 sec)

root@localhost:(none) 02:58:19 >select 8589934592/1024/1024/1024;
+-----+
| 8589934592/1024/1024/1024 |
+-----+
|          8.0000000000000 |
+-----+
1 row in set (0.00 sec)
```

说明：在MySQL数据库中进行查询发现，innodb_buffer_pool_size只有8G大小，对比参数文件中的48

4. 经过重启MySQL验证后，问题依然存在。

MySQL8.0直方图功能简介	435浏览
2025-03-21	
MySQL8.0统计信息总结	802浏览
2025-03-17	
MySQL8.0分区表之范围分区	290浏览
2025-01-24	
MySQL8.0新特性-通用表达式WITH	398浏览
2024-12-13	
MySQL未提交事务导致的TRUNCATE表阻塞挂起问题处理	781浏览
2024-12-13	

目录

- 适用范围
- 问题概述
- 问题分析
- MySQL8.0的持久化参数相关知识点：
- 解决方案
- 总结
- 参考文档

```
root@localhost:(none) 03:05:06 >shutdown;
Query OK, 0 rows affected (0.08 sec)

root@localhost:(none) 03:05:09 >exit
Bye
[root@iZf8zcal7db1wnlk0w6a8lZ ~]# /usr/local/mysql/bin/mysqld_safe --user=mysql
[1] 117682
[root@iZf8zcal7db1wnlk0w6a8lZ ~]# 2024-04-01T07:05:33.358753Z mysqld_safe Loggi
2024-04-01T07:05:33.399030Z mysqld_safe Starting mysqld daemon with databases f

[root@iZf8zcal7db1wnlk0w6a8lZ ~]#
[root@iZf8zcal7db1wnlk0w6a8lZ ~]# /usr/local/mysql/bin/mysql -uroot -p
Enter password:
Welcome to the MySQL monitor.  Commands end with ; or \g.
Your MySQL connection id is 193
Server version: 8.0.36 MySQL Community Server - GPL

Copyright (c) 2000, 2024, Oracle and/or its affiliates.

Oracle is a registered trademark of Oracle Corporation and/or its
affiliates. Other names may be trademarks of their respective
owners.

Type 'help;' or '\h' for help. Type '\c' to clear the current input statement.

root@localhost:(none) 03:07:00 >\s
-----
/usr/local/mysql/bin/mysql  Ver 8.0.36 for Linux on x86_64 (MySQL Community Ser

Connection id:          8
Current database:
Current user:           root@localhost
SSL:                   Not in use
Current pager:          stdout
Using outfile:          ''
Using delimiter:        ;
Server version:         8.0.36 MySQL Community Server - GPL
Protocol version:       10
Connection:             Localhost via UNIX socket
Server characterset:    utf8mb4
Db      characterset:   utf8mb4
Client characterset:    utf8mb4
Conn.  characterset:    utf8mb4
UNIX socket:            /data/mysql8036/3306/run/mysql.sock
Binary data as:         Hexadecimal
Uptime:                 1 min 28 sec

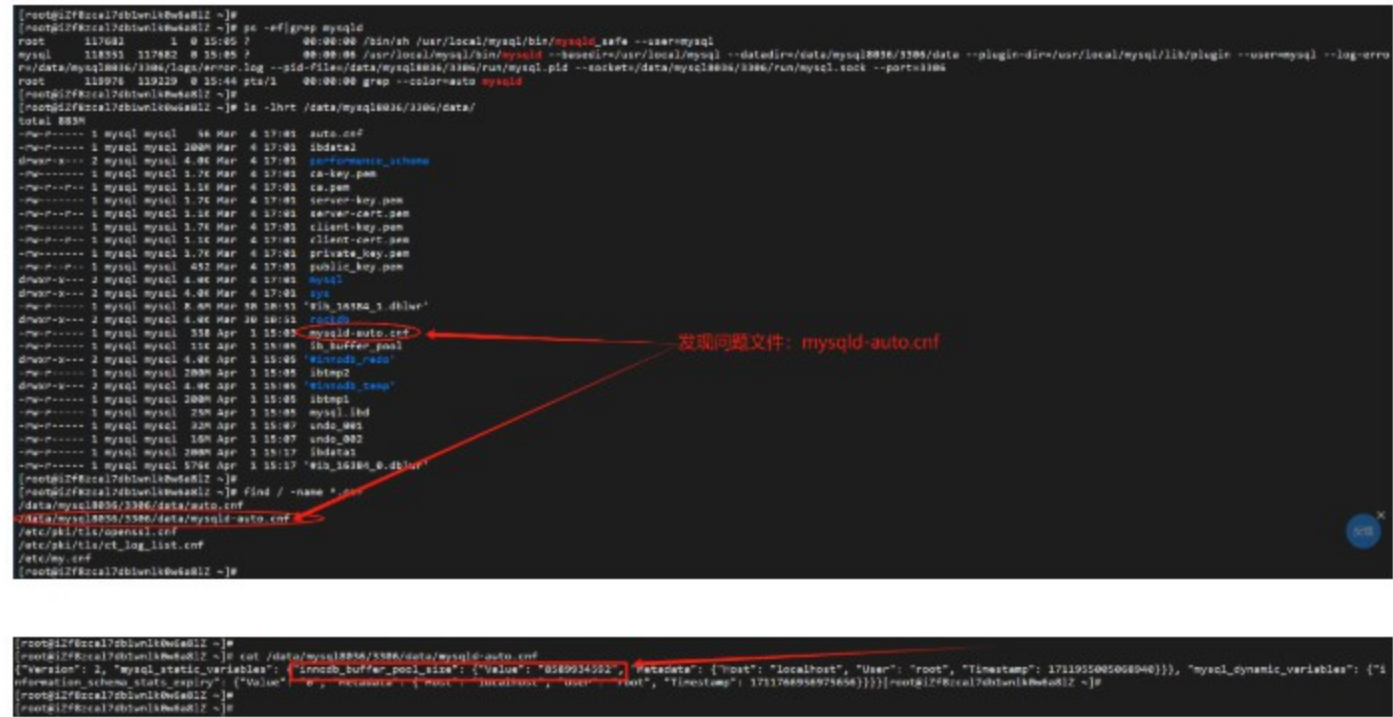
Threads: 2  Questions: 8  Slow queries: 0  Opens: 136  Flush tables: 3  Open ta
-----

root@localhost:(none) 03:07:01 >show global variables like '%innodb_buffer_pool
+-----+
| Variable_name          | Value          |
+-----+
| innodb_buffer_pool_size | 8589934592    |
+-----+
1 row in set (0.00 sec)
说明：MySQL实例重启后，innodb_buffer_pool_size大小依然为8G，和参数文件中指定的48G不匹配！
```


5. 发现问题：mysqld-auto.cnf文件

由于没有其他异常发现，但仍然随机去找可能得问题点（进程，目录结构等）看是否有其他异常！

```
[root@iZf8zcal7db1wnlk0w6a8lZ ~]# ps -eflgrep mysqld
root      117682      1  0 15:05 ?          00:00:00 /bin/sh /usr/local/mysql/bi
mysql     118351    117682  0 15:05 ?          00:00:06 /usr/local/mysql/bin/mysqld
root      119976    119229  0 15:44 pts/1     00:00:00 grep --color=auto mysqld
[root@iZf8zcal7db1wnlk0w6a8lZ ~]#
[root@iZf8zcal7db1wnlk0w6a8lZ ~]# ls -lhrt /data/mysql8036/3306/data/
total 883M
-rw-r----- 1 mysql mysql  56 Mar  4 17:01 auto.cnf
-rw-r----- 1 mysql mysql 200M Mar  4 17:01 ibdata2
drwxr-x--- 2 mysql mysql 4.0K Mar  4 17:01 performance_schema
-rw----- 1 mysql mysql 1.7K Mar  4 17:01 ca-key.pem
-rw-r--r-- 1 mysql mysql 1.1K Mar  4 17:01 ca.pem
-rw----- 1 mysql mysql 1.7K Mar  4 17:01 server-key.pem
-rw-r--r-- 1 mysql mysql 1.1K Mar  4 17:01 server-cert.pem
-rw----- 1 mysql mysql 1.7K Mar  4 17:01 client-key.pem
-rw-r--r-- 1 mysql mysql 1.1K Mar  4 17:01 client-cert.pem
-rw----- 1 mysql mysql 1.7K Mar  4 17:01 private_key.pem
-rw-r--r-- 1 mysql mysql  452 Mar  4 17:01 public_key.pem
drwxr-x--- 2 mysql mysql 4.0K Mar  4 17:01 mysql
drwxr-x--- 2 mysql mysql 4.0K Mar  4 17:01 sys
-rw-r----- 1 mysql mysql 8.6M Mar 30 10:51 '#ib_16384_1.dblwr'
drwxr-x--- 2 mysql mysql 4.0K Mar 30 10:51 rockdb
-rw-r----- 1 mysql mysql  338 Apr  1 15:03 mysqld-auto.cnf
-rw-r----- 1 mysql mysql  11K Apr  1 15:05 ib_buffer_pool
drwxr-x--- 2 mysql mysql 4.0K Apr  1 15:05 '#innodb_redo'
-rw-r----- 1 mysql mysql 200M Apr  1 15:05 ibtmp2
drwxr-x--- 2 mysql mysql 4.0K Apr  1 15:05 '#innodb_temp'
-rw-r----- 1 mysql mysql 200M Apr  1 15:05 ibtmp1
-rw-r----- 1 mysql mysql  25M Apr  1 15:05 mysql.ibd
-rw-r----- 1 mysql mysql  32M Apr  1 15:07 undo_001
-rw-r----- 1 mysql mysql  16M Apr  1 15:07 undo_002
-rw-r----- 1 mysql mysql 200M Apr  1 15:17 ibdata1
-rw-r----- 1 mysql mysql 576K Apr  1 15:17 '#ib_16384_0.dblwr'
[root@iZf8zcal7db1wnlk0w6a8lZ ~]#
[root@iZf8zcal7db1wnlk0w6a8lZ ~]# find / -name *.cnf
/data/mysql8036/3306/data/auto.cnf
/data/mysql8036/3306/data/mysqld-auto.cnf
/etc/pki/tls/openssl.cnf
/etc/pki/tls/ct_log_list.cnf
/etc/my.cnf
[root@iZf8zcal7db1wnlk0w6a8lZ ~]#
```



说明：从以上随机查找中，发现了异常文件mysqld-auto.cnf 终于恍然大悟，想起了MySQL8.0的持久化参数的特性。

MySQL8.0的持久化参数相关知识点：

SET PERSIST命令：

MySQL 8.0版本有一个新特性，在数据库中，引入了一个新命令 SET PERSIST，它允许用户将会话基本的SET PERSIST 语法会在datadir目录下创建一个json格式的mysqld-auto.cnf文件，在数据库重启后：如果 mysqld-auto.cnf文件中的参数与my.cnf文件中参数值重复，会覆盖my.cnf中参数的值！

持久化参数查询：

```
root@localhost:(none) 04:12:14 >
root@localhost:(none) 04:12:15 >select * from performance_schema.persisted_vari
+-----+-----+
| VARIABLE_NAME | VARIABLE_VALUE |
+-----+-----+
| information_schema_stats_expiry | 0 |
| innodb_buffer_pool_size | 8589934592 |
+-----+-----+
2 rows in set (0.00 sec)

root@localhost:(none) 04:12:17 >exit
Bye

[root@iZf8zcal7db1wnlk0w6a8lZ ~]# cat /data/mysql8036/3306/data/mysqld-auto.cnf
{"Version": 2, "mysql_static_variables": {"innodb_buffer_pool_size": {"Value":
[root@iZf8zcal7db1wnlk0w6a8lZ ~]#
```

```
root@localhost:(none) 04:12:14 >
root@localhost:(none) 04:12:15 >select * from performance_schema.persisted_variables;
+-----+-----+
| VARIABLE_NAME | VARIABLE_VALUE |
+-----+-----+
| information_schema_stats_expiry | 0 |
| innodb_buffer_pool_size | 8589934592 |
+-----+-----+
2 rows in set (0.00 sec)
```

解决方案

通过**RESET**命令将持久化参数文件中的设置清空！

```
[root@iZf8zcal7db1wnlk0w6a8lZ ~]#
[root@iZf8zcal7db1wnlk0w6a8lZ ~]# /usr/local/mysql/bin/mysql -uroot -p
Enter password:
Welcome to the MySQL monitor.  Commands end with ; or \g.
Your MySQL connection id is 11
Server version: 8.0.36 MySQL Community Server - GPL

Copyright (c) 2000, 2024, Oracle and/or its affiliates.

Oracle is a registered trademark of Oracle Corporation and/or its
affiliates. Other names may be trademarks of their respective
owners.

Type 'help;' or '\h' for help. Type '\c' to clear the current input statement.

root@localhost:(none) 04:16:16 >reset persist innodb_buffer_pool_size;
Query OK, 0 rows affected (0.00 sec)

root@localhost:(none) 04:16:38 >
root@localhost:(none) 04:16:38 >select * from performance_schema.persisted_vari
+-----+-----+
| VARIABLE_NAME          | VARIABLE_VALUE |
+-----+-----+
| information_schema_stats_expiry | 0              |
+-----+-----+
1 row in set (0.00 sec)

root@localhost:(none) 04:16:41 >system cat /etc/my.cnf|grep -i pool_size
innodb_buffer_pool_size          = 48G
root@localhost:(none) 04:17:35 >
root@localhost:(none) 04:17:36 >
```

说明：

reset persist <参数名>; 将清空mysqld-auto.cnf指定参数的配置，my.cnf文件中的配置不变！

reset persist; 将清空mysqld-auto.cnf文件中所有参数的配置，my.cnf文件中的配置不变！


```
root@localhost:(none) 04:38:21 >select * from performance_schema.persisted_vari
+-----+-----+
| VARIABLE_NAME          | VARIABLE_VALUE |
+-----+-----+
| information_schema_stats_expiry | 0              |
| max_connections         | 500            |
+-----+-----+
2 rows in set (0.00 sec)

root@localhost:(none) 04:38:36 >reset persist;
Query OK, 0 rows affected (0.00 sec)

root@localhost:(none) 04:39:09 >select * from performance_schema.persisted_vari
Empty set (0.00 sec)

root@localhost:(none) 04:39:12 >
root@localhost:(none) 04:39:13 >system cat /data/mysql8036/3306/data/mysqld-aut
{"Version": 2}root@localhost:(none) 04:39:48 >
root@localhost:(none) 04:39:49 >
```

总结

1. MySQL参数可以在运行时通过 SET GLOBAL 命令来更改，但是这种更改只会临时生效直到下次启动时数据库又会从配置文件（my.cnf）中读取。
2. MySQL8.0新增了 SET PERSIST命令，参数生效的同时，MySQL会将该命令的配置保存到数据目录下的 mysqld-auto.cnf 文件中，下次启动时会读取该文件用其中的配置来覆盖缺省的配置文件(my.cnf)。
3. 对于已经持久化了的变量（set persist），可以通过 reset persist 命令清除掉，这里只是清空 mysqld-auto.cnf 和 performance_schema.persisted_variables 中的内容，对于已经修改了的变量的值，不会产生影响。

参考文档

https://dev.mysql.com/doc/refman/8.0/en/persisted-system-variables.html

🔗 墨力计划 mysql8.0新特性 mysql mysql

最后修改时间：2024-04-02 17:37:17

「喜欢这篇文章，您的关注和赞赏是给作者最好的鼓励」

关注作者

赞赏

4人已赞赏



【版权声明】本文为墨天轮用户原创内容，转载时必须标注文章的来源（墨天轮），文章链接，文章作者等基本信息，否则作者和墨天轮有权追究责任。如果您发现墨天轮中有涉嫌抄袭或者侵权的内容，欢迎发送邮件至：contact@modb.pro进行举报，并提供相关证据，一经查实，墨天轮将立刻删除相关内容。

文章被以下合辑收录



关于MySQL的那些事儿（共39篇）

记录MySQL数据库的点点滴滴

收藏合集

评论

分享你的看法，一起交流吧～

- 

nick_liu

LV.3

牛逼，要是不知道8.0版本的新特性，是真的很难找出问题呢！

1年前

点赞

评论
- 

11 0

LV.5



1年前

点赞

评论
- 

百分

LV.7

DBA并不会消失，只会以新的方式出现，要充分掌握一些新的技能。

1年前

点赞

评论
- 

lynn

LV.1

学习！

1年前

点赞

评论
- 

天空之城

LV.1

学习！

1年前

1

评论
- 

手机用户0069

LV.1



1年前

点赞

评论

查看更多 >

相关阅读

- ACDU周度精选 | 本周数据库圈热点 + 技术干货分享（2025/7/25期）

墨天轮小助手

473次阅读

2025-07-25 15:54:18
- ACDU周度精选 | 本周数据库圈热点 + 技术干货分享（2025/7/17期）

墨天轮小助手

437次阅读

2025-07-17 15:31:18
- 墨天轮「实操看我的」数据库主题征文活动启动

墨天轮编辑部

390次阅读

2025-07-22 16:11:27
- 深度解析MySQL的半连接转换

听见风的声音

207次阅读

2025-07-14 10:23:00
- MySQL 9.4.0 正式发布，支持 RHEL 10 和 Oracle Linux 10

严少安

206次阅读

2025-07-23 01:21:32
- 索引条件下推和分区—一条SQL语句执行计划的分析

听见风的声音

198次阅读

2025-07-23 09:22:58

null和子查询--not in和not exists怎么选择？

听见风的声音 182次阅读 2025-07-21 08:54:19

MySQL数据库SQL优化案例(走错索引)

陈举超 167次阅读 2025-07-17 21:24:40

使用 MySQL Clone 插件为MGR集群添加节点

黄山谷 165次阅读 2025-07-23 22:04:19

MySQL 8.0.40：字符集革命、窗口函数效能与DDL原子性实践

shunwah  141次阅读 2025-07-15 15:27:19