**Ctrip MySQL审计使用手册**

**上海携程旅游信息技术有限公司**

**技术研发中心 DBA组**

2015年2月

# **文档版本**

|  |  |  |  |
| --- | --- | --- | --- |
| **版本号** | **编写人** | **编写日期** | **更改内容** |
| 1.0 | 姜宇祥 | 2015-2-15 | 创建文档 |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |

# **目录**

[**文档版本** 1](#_Toc414874608)

[1.0 2](#_Toc414874609)

[**目录** 3](#_Toc414874610)

[**一、引言** 4](#_Toc414874611)

[1编写目的 4](#_Toc414874612)

[2参考资料 4](#_Toc414874613)

[3术语和缩写词 4](#_Toc414874614)

[二、概述 5](#_Toc414874615)

[1、用途 5](#_Toc414874616)

[2、运行环境 5](#_Toc414874617)

[3、性能 5](#_Toc414874618)

[三、使用 6](#_Toc414874619)

[1、构成 6](#_Toc414874620)

[2、安装 6](#_Toc414874621)

[2-1加载插件 6](#_Toc414874622)

[2-2卸载插件 6](#_Toc414874623)

[3、使用配置文件 6](#_Toc414874624)

[3-1、配置文件的格式 6](#_Toc414874625)

[3-2、配置 7](#_Toc414874626)

[4、运行时变量 8](#_Toc414874627)

[4-1、变量 8](#_Toc414874628)

[4-2、变量说明 9](#_Toc414874629)

[5、运行时状态 11](#_Toc414874630)

[6、SQL命令 11](#_Toc414874631)

[6、输出 12](#_Toc414874632)

[7、异常 12](#_Toc414874633)

# **一、引言**

## 1编写目的

编写本使用说明的目的是充分叙述本软件所能实现的功能及其运行环境，以便使用者了解本软件的使用范围和使用方法，并为软件的维护和更新提供必要的信息。

## 2参考资料

1、MySQL Enterprise Audit Log Plugin

http://dev.mysql.com/doc/refman/5.5/en/audit-log-plugin.html

## 3术语和缩写词

审计插件：如无特殊说明，特指携程的mysql审计插件

# 二、概述

## 1、用途

用于mysql数据库的事件审计

## 2、运行环境

本软件作为插件运行在携程定制的mysql5.6.21版本上，mysql的工作环境为centos5/6

## 3、性能

在全部审计功能打开的情况下，mysql性能下降不超过10%

# 三、使用

## 1、构成

该软件随携程mysql安装包共同发布。安装后包括动态链接库ctrip\_audit.so和配置文件ctrip\_audit.cnf两个部分

## 2、安装

### 2-1加载插件

install plugin ctrip\_audit soname 'ctrip\_audit.so';

### 2-2卸载插件

uninstall plugin ctrip\_audit;

## 3、使用配置文件

携程mysql审计插件使用独立的配置文件，并不从mysql的配置文件读取配置信息。通过读取配置文件获得运行时所需信息，并根据这些配置信息完成相应的审计功能。

审计插件的配置文件需要与插件放在相同目录，也就是mysql的插件目录中。如果没有找到配置文件，审计插件将按照默认模式进行工作。

配置文件由通用组和过滤组构成。通用组用于配置审计插件的通用属性，过滤组用于配置运行时的过滤规则，每一个过滤组对应一个过滤规则。

### 3-1、配置文件的格式

审计插件是按照行进行配置读取的，每一行代表一个完整的内容，不支持分行的配置内容读取，每一行称为配置行。

配置行分为如下4类，配置组名，配置项，注释和空行。其中注释是以“#”字符开头的行，空行是包含空格/制表符/回车/换行字符，这两种行将被审计插件忽略。配置组名和配置项将在后面说明。

### 3-2、配置

审计插件支持两种配置组。通用配置组和过滤配置组。

通用配置组须以[general]为内容的配置行开始，其后的为通用配置的配置项。

过滤配置组须以[audit rule]为内容的配置行开始，其后为过滤组的配置内容。

#### 3-2-1通用配置组

通用配置组支持的配置为

log\_file:用于记录审计信息

err\_file:用于记录用户请求执行时出现的错误

enable\_buffer:审计时是否使用缓冲区。使用缓冲区将会提高程序的运行效率，但发生宕机情况时会出现部分数据未写入磁盘，导致部分审计信息丢失的情况

#### 3-2-2过滤配置组

过滤配置组支持的配置为

name:过滤配置的名字

user:进行过滤的用户名字

host:进行过滤的主机名或ip

event:进行过滤的事件。携程审计的事件分为两大类，连接审计事件和普通审计事件

连接事件包括连接/断连/改变用户

普通事件包括状态/错误

#### 3-2-3书写要求

1、除空配置行外，其他配置行不接受以空格为开始的配置行。

2、配置组/配置项不区分大小写。

3、配置项以key-value方式书写，’=’字符作为分隔符，前端为key，后端为value，’=’两端的空格也作为key和value的输入

4、过滤组中，用户/主机名配置项中value允许使用的字符按照如下正则表达式表示[a\_z][A\_Z][0\_9][\_.]

#### 3-2-5样例

[general]

audit\_file=ctrip\_audit.log

audit\_error\_file=ctrip\_audit\_error.log

enable\_buffer=1

[audit rule]

name=rule1

user=root

host=192.168

event=connection:connect;connection:disconnect;general:status;general:error

## 4、运行时变量

通过查询一些运行时变量，用户可以了解审计运行中的参数；通过设置一些运行时变量，改变审计运行中的参数，影响审计的行为。

### 4-1、变量

审计插件所使用的变量如下表所列

|  |  |  |
| --- | --- | --- |
| 变量名 | 说明 | 运行时读写 |
| ctrip\_audit\_add\_rule | 增加审计条件 | 读/写 |
| ctrip\_audit\_buffer\_size | 缓冲区大小 | 读/写 |
| ctrip\_audit\_enable\_buffer | 打开缓冲区 | 读/写 |
| trip\_audit\_error\_log\_file | SQL错误审计文件 | 只读 |
| ctrip\_audit\_flush\_log | 刷新日志 | 读/写 |
| ctrip\_audit\_log\_file | 审计文件 | 只读 |
| ctrip\_audit\_remove\_rule | 移除审计条件 | 读/写 |
| ctrip\_audit\_rules | 查看审计条件 | 只读 |

### 4-2、变量说明

#### 4-2-1、ctrip\_audit\_add\_rule

通过设置该变量，增加审计过滤条件

类型:

字符类型

样例:

set global ctrip\_audit\_add\_rule='name=n1;host=localhost;user=user1';

#### 4-2-2、ctrip\_audit\_buffer\_size

设置缓冲区的大小，以KB为单位进行设置。如1表示1Kb，也就是1024个字节

类型

整数类型

样例

set global ctrip\_audit\_buffer\_size=32;

#### 4-2-3、ctrip\_audit\_enable\_buffer

打开审计缓冲功能

类型

布尔类型

样例

set global ctrip\_audit\_enable\_buffer=true;

#### 4-2-4、ctrip\_audit\_error\_log\_file

保存SQL错误的审计文件

类型

字符类型

样例

只读变量，只能通过show variable得到当前的信息

#### 4-2-5、ctrip\_audit\_flush\_log

刷新当前的的审计文件，旧的数据将从当前的文件中删除，保存在其他文件中。新的数据继续写入当前的文件。

该变量的值只能被设置为true，用以表示进行审计信息的刷新

类型

布尔类型

样例

set global ctrip\_audit\_flush\_log=true;

#### 4-2-6、ctrip\_audit\_log\_file

保存mysql的审计信息内容。

类型

布尔类型

样例

只读变量，只能通过show variable得到当前的信息

#### 4-2-7、ctrip\_audit\_remove\_rule

移除审计的过滤条件。通过制定过滤条件的名字，进行过滤条件的删除

类型

字符类型

样例

set global ctrip\_audit\_remove\_rule='name=n1';

#### 4-2-8、ctrip\_audit\_rules

查看当前的审计过滤条件

类型

字符类型

样例

只读变量，只能通过show variable得到当前的信息

## 5、运行时状态

通过查询运行时的状态可以了解携程审计插件的运行状态，支持的状态包括如下。

|  |  |
| --- | --- |
| 状态变量 | 说明 |
| Ctrip\_audit\_called | 调用审计插件的次数 |
| Ctrip\_audit\_connection\_change\_user | 传入connection change user事件的次数 |
| Ctrip\_audit\_connection\_connect | 传入connection connect事件的次数 |
| Ctrip\_audit\_connection\_disconnect | 传入connection disconnect事件的次数 |
| Ctrip\_audit\_general\_error | 传入general error事件的次数 |
| Ctrip\_audit\_general\_log | 传入general log事件的次数 |
| Ctrip\_audit\_general\_result | 传入general result事件的次数 |
| Ctrip\_audit\_general\_status | 传入general status事件的次数 |

说明：

1、general log/general result事件在携程审计插件中是不进行审计的

## 6、SQL命令

命令

flush logs "any string";

说明

与设置变量ctrip\_audit\_flush\_log产生的效果相同

## 7、输出

审计事件以json的数据格式输出到审计文件中。如下样例所示

{

"version": "1.1",

"timestamp": "2015-03-20 09:27:42",

"type": "general",

"user": "root[root] @ localhost []",

"host": "localhost",

"ip": "",

"command\_class": "set\_option",

"sqltext": "set global ctrip\_audit\_flush\_log=1"

}

## 8、异常

1、在读取分析配置文件中如果出现错误，插件将会返回错误给mysql，并停止加载

2、在设置运行时变量未能成功的情况下，报告错误，并保持设置命令之前的状态

3、连接在特定审计失效。在连接非常活跃的情况，由于mysql的实现机制会导致连接的资源无法被正确释放。

如下是mysql出现问题的代码。红色标注部分是错误的核心，在插件卸载后，连接的工作线程执行到该函数的红色部分跳出，导致后续的释放代码没有正确执行。其中标志位没有重置（蓝色标注部分），导致下次进入审计时，无法正确的判断连接状态。

void mysql\_audit\_release(THD \*thd)

{

plugin\_ref \*plugins, \*plugins\_last;

if (!thd || !(thd->audit\_class\_plugins.elements))

return;

plugins= (plugin\_ref\*) thd->audit\_class\_plugins.buffer;

plugins\_last= plugins + thd->audit\_class\_plugins.elements;

for (; plugins < plugins\_last; plugins++)

{

st\_mysql\_audit \*data= plugin\_data(\*plugins, struct st\_mysql\_audit \*);

/\* Check to see if the plugin has a release method \*/

if (!(data->release\_thd))

continue;

/\* Tell the plugin to release its resources \*/

data->release\_thd(thd);

}

/\* Now we actually unlock the plugins \*/

plugin\_unlock\_list(NULL, (plugin\_ref\*) thd->audit\_class\_plugins.buffer,

thd->audit\_class\_plugins.elements);

/\* Reset the state of thread values \*/

reset\_dynamic(&thd->audit\_class\_plugins);

memset(thd->audit\_class\_mask, 0, sizeof(thd->audit\_class\_mask));

}