

社区首页 > 专栏 > MySQL proxysql 在深入 信息获取和信息输出

MySQL proxysql 在深入 信息获取和信息输出

发布于 2020-09-10 16:14:18 1.7K 0 举报

文章被收录于专栏: AustinDatabases

关联问题

换一批

MySQL ProxySQL如何实现...

MySQL ProxySQL怎样进行...

MySQL ProxySQL信息获取...



ProxySQL 前两期针对不同的MySQL高可用的方式来说的, 本期会更加深入到PROXYSQL 的一些本身的东西. 在PROXYSQL 中有一个表,存储了所有的global variables, 其中包含了大量的变量值.实际上,这些就是键值对. 其中分为 MySQL 为前缀的和 以 admin为前缀的两个部分.

```
Admin> select * from global_variables;
+-----+-----+
| variable_name | variable_value |
+-----+-----+
```


mysql-default_action	
mysql-default_names	
mysql-default_sql_safe_updates	
mysql-default_sql_select_limit	
mysql-default_sql_mode	
mysql-default_time_zone	
mysql-default_isolation_level	
mysql-default_transaction_read	
mysql-default_sql_auto_is_null	
mysql-default_net_write_timeout	
mysql-default_max_join_size	
mysql-default_sql_log_bin	
mysql-default_wsrep_sync_wait	
mysql-default_group_concat_max_len	
mysql-shun_on_failures	5
mysql-shun_recovery_time_sec	10
mysql-query_retries_on_failure	1
mysql-client_multi_statements	true
mysql-connect_retries_delay	1
mysql-connection_delay_multiplex_ms	0
mysql-connection_max_age_ms	0
mysql-connect_timeout_server_max	10000
mysql-eventslog_filename	
mysql-eventslog_filesize	104857600
mysql-eventslog_default_log	0
mysql-eventslog_format	1
mysql-auditlog_filename	
mysql-auditlog_filesize	104857600
mysql-handle_unknown_charset	1
mysql-free_connections_pct	10
mysql-connection_warming	false
mysql-session_idle_ms	1000
mysql-have_ssl	false
mysql-client_found_rows	true
mysql-monitor_enabled	true
mysql-monitor_connect_timeout	600
mysql-monitor_ping_max_failures	3
mysql-monitor_ping_timeout	1000
mysql-monitor_read_only_max_timeout_count	3
mysql-monitor_replication_lag_interval	10000
mysql-monitor_replication_lag_timeout	1000
mysql-monitor_groupreplication_healthcheck_interval	5000
mysql-monitor_groupreplication_healthcheck_timeout	800
mysql-monitor_groupreplication_healthcheck_max_timeout_count	3
mysql-monitor_groupreplication_max_transactions_behind_count	3
mysql-monitor_groupreplication_healthcheck_interval	5000

其中mysql 为前缀的主要是与MYSQL 有关的变量,

1 mysql-auditlog_filename 这个变量主要是建立一个auditlog , 给出相关路径和文件名, 会在指定的位置生成 audit log , audit log 主要是记录与 中间件[®] proxysql 连接的账号,地址以及操作的 数据库[®] 对象, 使用json的格式.

update global_variables set variable_value = '/proxysql/log/audit.log' where variable_name = 'mysql-auditlog_filename' ;

```
[proxysql@proxysql log]$ cat audit.log.00000001
{"client_addr":"192.168.198.106:34334","event":"MySQL_Client_Connect_OK","proxy_addr":"0.0.0.0:6033","schemaname":"mysql","ssl":false,"thread_id":131,"time":"2020-05-19 23:34:50.061","timestamp":1589945690061,"username":"contact"}
{"client_addr":"192.168.198.106:34334","event":"MySQL_Client_Init_DB","proxy_addr":"0.0.0.0:6033","schemaname":"test","ssl":false,"thread_id":131,"time":"2020-05-19 23:34:56.614","timestamp":1589945698814,"username":"contact"}
[proxysql@proxysql log]$
```

其中包含访问的时间,数据库账号,访问的源地址等等信息,基本上

2 记录访问MYSQL中的一些访问的语句通过不同的条件来过滤

从proxysql 2.0.6 开始可以记录查询中的相关的记录, 默认这个开关是关闭的


```
Admin> select * from global_variables where variable_name = 'mysql-eventslog_filename';
+-----+-----+
| variable_name | variable_value |
+-----+-----+
| mysql-eventslog_filename | queries.log |
+-----+-----+
1 row in set (0.00 sec)
```

微信号: AustinDatabases

配置是相对简单的,例如我们对某个应用账号的执行的语句进行监控,则我们可以做以下工作.

```
SET mysql-eventslog_filename='queries.log';

SET mysql-eventslog_default_log=1;

INSERT INTO mysql_query_rules (rule_id, active,username, log,apply) VALUES (1,1,'contact',1,0);

SET mysql-eventslog_format=2;

SAVE MYSQL VARIABLES TO DISK;

LOAD MYSQL VARIABLES TO RUNTIME;
```

在操作完毕后,凡是 contact 账号操作的语句会被记录在queries.log 文件中

3 关于proxysql 的工作情况

虽然PROXYSQL 比较皮实耐用,但了解内部的系统的一些运行状态还是重要的,

```
Admin>
Admin> SELECT * FROM stats.stats_memory_metrics;
+-----+-----+
| Variable_Name | Variable_Value |
+-----+-----+
| SQLite3_memory_bytes | 3290280 |
| jemalloc_resident | 16416768 |
| jemalloc_active | 13852672 |
| jemalloc_allocated | 13104536 |
| jemalloc_mapped | 20144128 |
| jemalloc_metadata | 3164776 |
| jemalloc_retained | 1875968 |
| Auth_memory | 702 |
| query_digest_memory | 0 |
| mysql_query_rules_memory | 2580 |
| mysql_firewall_users_table | 0 |
| mysql_firewall_users_config | 0 |
| mysql_firewall_rules_table | 0 |
| mysql_firewall_rules_config | 329 |
| stack_memory_mysql_threads | 67108864 |
| stack_memory_admin_threads | 8388608 |
| stack_memory_cluster_threads | 0 |
+-----+-----+
17 rows in set (0.00 sec)

Admin> █
```

微信号: AustinDatabases

- jemalloc_active 已经分配给应用的页面
- jemalloc_allocated 已经分配给应用的字节
- jemalloc_metadata 元数据库分配的内存
- Auth_memory 认证模块分配的内存
- SQLite3_memory_bytes PROXYSQL 分配的内存
- query_digest_memory 上面刚刚提到的,对执行的语句进行过滤分配的内存
- ProxySQL 本身也对执行的语句进行一个统计,下面的


```
Admin> select * from global_variables where variable_name = 'mysql-commands_stats';
+-----+-----+
| variable_name | variable_value |
+-----+-----+
| mysql-commands_stats | true |
+-----+-----+
1 row in set (0.00 sec)
```

微信号: AustinDatabases

select * from global_variables where variable_name = 'mysql-commands_stats';

默认统计PROXYSQL 接受的语句,并根据语句的反馈进行相关的统计

SELECT * FROM stats_mysql_commands_counters;

可以根据其中的已有的统计项对相关的命令进行一个统计, 这弥补了部分MYSQL方面的这部分的统计数据获取的麻烦.

```
Admin> select * from stats_mysql_connection_pool;
+-----+-----+-----+-----+-----+-----+-----+-----+-----+-----+-----+-----+-----+-----+
| hostgroup | srv_host | srv_port | status | ConnUsed | ConnFree | ConnOK | ConnERR | MaxConnUsed | Queries | Queries_GTID_sync | Bytes_data_sent | Bytes_data_recv | Latency_us |
+-----+-----+-----+-----+-----+-----+-----+-----+-----+-----+-----+-----+-----+-----+
| 600 | 192.168.198.102 | 3306 | ONLINE | 0 | 1 | 1 | 0 | 1 | 7 | 0 | 117 | 182 | 1199 |
| 600 | 192.168.198.101 | 3306 | ONLINE | 0 | 0 | 0 | 6 | 1 | 0 | 0 | 0 | 0 | 1340 |
| 601 | 192.168.198.102 | 3306 | ONLINE | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 1199 |
| 601 | 192.168.198.101 | 3306 | ONLINE | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 1340 |
+-----+-----+-----+-----+-----+-----+-----+-----+-----+-----+-----+-----+-----+-----+
4 rows in set (0.00 sec)
```

微信号: AustinDatabases

上图是通过对 stats_mysql_connection_pool 进行访问,得到当前外部访问MYSQL 的状态, 这个表可以显现,当前有多少连接针对指定的 数据库服务 器,并且产生的queries ,产生的数据发送bytes ,接受到的bytes 等等信息

```
Admin> select * from stats_mysql_processlist;
+-----+-----+-----+-----+-----+-----+-----+-----+-----+-----+-----+-----+-----+-----+
| ThreadID | SessionID | user | db | cli_host | cli_port | hostgroup | l_srv_host | l_srv_port | srv_host | srv_port | command | time_ms | info | status_flags | extended_info |
+-----+-----+-----+-----+-----+-----+-----+-----+-----+-----+-----+-----+-----+-----+
| 4 | 2 | contact | mytest | 192.168.198.106 | 36576 | 600 | NULL | NULL | NULL | NULL | SHOW TABLES | 1 |  | 0 |  |
+-----+-----+-----+-----+-----+-----+-----+-----+-----+-----+-----+-----+-----+-----+
1 row in set (0.00 sec)
```

微信号: AustinDatabases

在MYSQL 中可以通过 show processlist 来查看当前和MYSQL正在交互的session的清空, PROXYSQL 中也是提供 stats_mysql_processlist 对系统当前连接进来的连接进行展示.

在MYSQL中5.7 已经具备了追踪历史查询的功能, proxysql 也提供了类似的功能并且统计的信息会更全面提取更方便.

select * from stats_mysql_query_digest;

```
Admin> select * from stats_mysql_query_digest;
+-----+-----+-----+-----+-----+-----+-----+-----+-----+-----+-----+-----+-----+-----+
| hostgroup | schemaname | username | client_address | digest | digest_text | count_star | first_seen | last_seen | sum_time | min_time | max_time | sum_rows_affected | sum_rows_sent |
+-----+-----+-----+-----+-----+-----+-----+-----+-----+-----+-----+-----+-----+-----+
| 600 | mytest | contact |  | 0x0B4AA150E94FB308 | SELECT * FROM mytest WHERE ?=? | 1 | 1589976401 | 1589976401 | 141991 | 141991 | 141991 | 0 | 0 |
| 600 | mytest | contact |  | 0x99531AEFF718C501 | show tables | 2 | 1589976401 | 1589976403 | 48232 | 23822 | 24410 | 0 | 2 |
| 600 | mytest | contact |  | 0x2C9CDE14087A21B | select * from mytest | 1 | 1589976409 | 1589976409 | 3906 | 3906 | 3906 | 0 | 0 |
| 600 | mytest | contact |  | 0x02033E45904D30F0 | show databases | 1 | 1589976401 | 1589976401 | 805 | 8205 | 8205 | 0 | 5 |
| 600 | mysql | contact |  | 0x6208320FE90D71A | SELECT DATABASE() | 1 | 1589976401 | 1589976401 | 0 | 0 | 0 | 0 | 0 |
| 600 | mysql | contact |  | 0x02033E45904D30F0 | show databases | 1 | 1589976395 | 1589976395 | 0 | 0 | 0 | 0 | 0 |
| 600 | mysql | contact |  | 0x226CD90D52A2BA0B | select @@version_comment limit 1 | 1 | 1589976381 | 1589976381 | 0 | 0 | 0 | 0 | 0 |
+-----+-----+-----+-----+-----+-----+-----+-----+-----+-----+-----+-----+-----+-----+
7 rows in set (0.00 sec)
```

微信号: AustinDatabases

或者统计当前的与数据库有关的账户连接的情况

```
Admin> select * from stats_mysql_users;
+-----+-----+-----+-----+
| username | frontend_connections | frontend_max_connections |
+-----+-----+-----+-----+
| contact | 1 | 10000 |
+-----+-----+-----+-----+
1 row in set (0.00 sec)
```

微信号: AustinDatabases

mysql 本身统计相关信息的方式也不少,但proxysql 将这些统计信息做的更好,并且提取时也不会和MYSQL 本身有冲突.

除此以外proxysql 也具备相关的丰富的日志记录功能,在数据库中有一个monitor 数据库,其中存储着相关的监控记录, 以下举几个例子


```
Admin> show tables from monitor;
+-----+
| tables |
+-----+
| mysql_server_aws_aurora_check_status |
| mysql_server_aws_aurora_failovers   |
| mysql_server_aws_aurora_log          |
| mysql_server_connect_log             |
| mysql_server_galera_log               |
| mysql_server_group_replication_log   |
| mysql_server_ping_log                 |
| mysql_server_read_only_log           |
| mysql_server_replication_lag_log     |
+-----+
9 rows in set (0.00 sec)
```

Admin>

微信号: AustinDatabases

1 如何判断proxysql 和 数据库 服务器之间的网络响应的问题

通过下面的语句

select * from mysql_server_ping_log where time_start_us in (select time_start_us from mysql_server_ping_log order by time_start_us desc limit 3);

```
MySQL [monitor]> select * from mysql_server_ping_log where time_start_us in (select time_start_us from mysql_server_ping_log order by time_start_us desc limit 3);
+-----+-----+-----+-----+-----+
| hostname | port | time_start_us | ping_success_time_us | ping_error |
+-----+-----+-----+-----+-----+
| 10.5.38.20 | 3306 | 1597720309594910 | 888 | NULL |
| 10.5.38.20 | 3306 | 1597720319442485 | 710 | NULL |
| 10.5.38.21 | 3306 | 1597720319582335 | 633 | NULL |
+-----+-----+-----+-----+-----+
3 rows in set (0.00 sec)
```

微信号: AustinDatabases

通过遍历最新的记录，可以判断数据库和PROXYSQL的连通性和时间延迟，或者根本就无法连接的问题。

2 PROXYSQL 与数据库连接之间的时间

select * from mysql_server_connect_log where time_start_us in (select time_start_us from mysql_server_connect_log order by time_start_us desc limit 3);

```
MySQL [monitor]> select * from mysql_server_connect_log where time_start_us in (select time_start_us from mysql_server_connect_log order by time_start_us desc limit 3);
+-----+-----+-----+-----+-----+
| hostname | port | time_start_us | connect_success_time_us | connect_error |
+-----+-----+-----+-----+-----+
| 10.5.38.21 | 3306 | 1597720751159709 | 2111 | NULL |
| 10.5.38.20 | 3306 | 1597720810118219 | 2172 | NULL |
| 10.5.38.21 | 3306 | 1597720810912871 | 2022 | NULL |
+-----+-----+-----+-----+-----+
3 rows in set (0.00 sec)
```

微信号: AustinDatabases

3 判断到底当前主库和从库之间到底谁是主谁是从。

select hostname,time_start_us,success_time_us,read_only from mysql_server_read_only_log where time_start_us in (select time_start_us from mysql_server_read_only_log order by time_start_us desc limit 5);

```
MySQL [monitor]> select * from mysql_server_connect_log where time_start_us in (select time_start_us from mysql_server_connect_log order by time_start_us desc limit 3);
+-----+-----+-----+-----+-----+
| hostname | port | time_start_us | connect_success_time_us | connect_error |
+-----+-----+-----+-----+-----+
| 10.5.38.21 | 3306 | 1597720751159709 | 2111 | NULL |
| 10.5.38.20 | 3306 | 1597720810118219 | 2172 | NULL |
| 10.5.38.21 | 3306 | 1597720810912871 | 2022 | NULL |
+-----+-----+-----+-----+-----+
3 rows in set (0.00 sec)
```

微信号: AustinDatabases

这三者有什么关系，1 最基本的是中间件和数据库之间是否连通，如果此处出现问题，则表明中间件和数据库之间的网络或者连通的数据库本身是否出现问题，出现问题就需要告警

2 此时与数据库连接之间的问题是1来判断，而2 则是判断PROXYSQL 与数据库之间的连接的状态的问题，这里主要注意的是，连接的时间

3 则是判定 数据流的目的地是哪里，整体的数据复制集合中，主库是否存在通过这个日志可以判断当前的数据流的流向以及主从库的状态问题。

所以完全可以通过PROXYSQL 来写一套程序来判断当前整体MYSQL复制集的状态。

另外本身PROXYSQL 的stats_ 就可以通过MYSQL 协议来进行各种状态的获取,并且一些开源监控软件 PMM 也可以对其进行监控. PROXYSQL 也对监控的方式提供了 WEB 方式，提供的数据比较有限,但在没有其他方式的加持和对此进行一个图形监控的时候,这个方式也是一个好方法.

ProxySQL

[Home](#) [System](#) [MySQL](#) [Query Cache](#)

Statistics:

[Last 1/2 hour](#) [Last 8 hours](#) [Last 7 days](#)
[Last 1 hour](#) [Last 1 day](#) [Last 1 month](#)
[Last 2 hours](#) [Last 3 days](#) [Last 3 months](#)



今天就到此为止,但PROXYSQL 本身的功能是强大了,例如读写分离, 以及控制读写分离在主从延迟后的如何操作都有相关的设置



本文参与 [腾讯云自媒体同步曝光计划](#)，分享自微信公众号。

原始发表：2020-09-04，如有侵权请联系 cloudcommunity@tencent.com 删除



网络安全

sql

数据库

云数据库 SQL Server

访问管理

评论



[登录](#) 后参与评论

推荐阅读

编辑精选文章

[换一批](#)

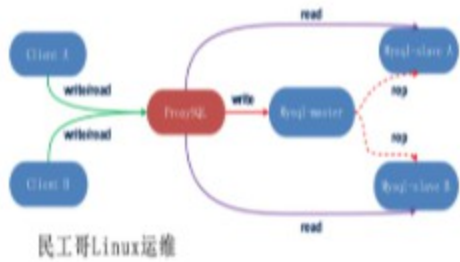
- | | | | |
|----------------------|------|--------------------|-------|
| 多租户的 4 种常用方案 | 4750 | 亿级月活的社交 APP，陌陌... | 3369 |
| 60页PPT全解：DeepSeek... | 4722 | Java与Go差别在哪，谁要被... | 4105 |
| 大模型 Token 究竟是啥：图... | 3211 | MCP协议详解：一文读懂跨... | 10417 |

ProxySQL+Mysql实现数据库读写分离实战

🔒 网络安全 sql 数据库 云数据库 SQL Server

ProxySQL是一个高性能的MySQL中间件，拥有强大的规则引擎。具有以下特性：
<http://www.proxysql.com/>

 民工哥 · 2020/09/16 👁 3.4K 💬 0



MySQL读写分离(ProxySQL)

🔒 云数据库 SQL Server https 网络安全 github git

读写分离就是用户在发送请求时，请求经过中间件，中间件将请求中的读和写操作分辨出来将读请求发送给后端的从服务器，将写请求发送给后端的主服务器，再又主...

 Java帮帮 · 2019/05/17 👁 6.4K 💬 0



数据库中间件ProxySQL读写自动分离实践

🔒 负载均衡 访问管理 云数据库 SQL Server sql

腾讯云cdb可以提供主库VIP+只读从库VIP的形式，这样开发使用时需要维护2个VIP而且不能自动读写分离，基于现状计划使用proxysql结合clb及cdb来实现读写自动分离。 架构如下： app--clb四层--proxysql--主vip+自...

 三杯水Plus · 2019/12/30 👁 639 💬 0

用 ProxySQL 实现 MySQL 的读写分离

🔒 mysql

这个配置包含了基本的故障转移设置。如果从服务器的复制延迟超过10秒，ProxySQL会将查询路由到其他可用的从服务器或主服务器。

 贺春旸的技术博客 · 2024/07/30 👁 250 💬 0

MySQL高可用实现：主从结构下ProxySQL中的读写分离

🔒 sqlite sql server android 网络安全 sql

ProxySQL是一个高性能的MySQL中间件，拥有强大的规则引擎。具有以下特性：

 数据和云01 · 2019/06/15 👁 945 💬 0

ProxySQL 入门教程

🔒 sqlite android 网络安全 sql 数据库

目前可以很好的支持 Master Slave\ MGR \ PXC等，并提供连接池、读写分离、日志记录等功能，当然还有很多其他实用功能，这里不一一列举了。

 iMike · 2019/07/10 👁 3.9K 💬 0

技术分享 | ProxySQL 搭配 MySQL HA （上）

sql 数据库 云数据库 SQL Server 网络安全

资深数据库专家，专研 MySQL 十余年。擅长 MySQL、PostgreSQL、MongoDB 等开源数据库相关的备份恢复、SQL 调优、监控运维、高可用架构设计等。目前任职于爱可生，为各大运营商及银行金融企业提供 MySQ...

 爱可生开源社区 · 2021/12/27 1K 0

万字详述 MySQL ProxySQL

https 网络安全 github git 开源

墨墨导读：Proxysql读写分离的中间件，支持高可用 主从\ MGR \ PXC等环境，并提供连接池、缓存、日志记录等功能。

 数据和云 · 2020/08/14 2.3K 0



ProxySQL Cluster 概述

sql server 网络安全 sql 数据库 云数据库 SQL Server

在 ProxySQL 1.4.2 之前，ProxySQL 单点的解决方法有配合 keepalived 使用来实现 ProxySQL 的主备，但是需要在主备上配置两份完全相同的路由或规则，如果再没有自动运维平台，同时维护两份配置的也是相当麻烦的。...

 阿炳数记 · 2019/02/27 995 0

MYSQL 中间件 为什么选择 PROXYSQL VS INNODB CLUSTER

sql 数据库 云数据库 SQL Server 云推荐引擎 数据分析

没有磨难，怎么能得到心灵的平静，吃过苦，才懂得甜的味道,Just waitting .

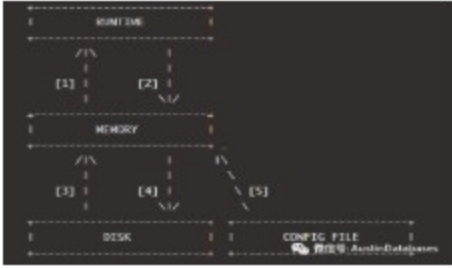
 AustinDatabases · 2020/08/27 706 0

MYSQL PROXYSQL的基本原理，配置，与query cache.

sql server 网络安全 缓存 云数据库 SQL Server sql

PorxySQL 作为使用MYSQL的一种中间件，为什么是目前比较流行的中间件之一，个人总结有以下几点。

 AustinDatabases · 2020/09/10 1.5K 0



【DB宝42】MySQL高可用架构MHA+ProxySQL实现读写分离和负载均衡

云数据库 SQL Server 容器 负载均衡 负载均衡缓存

之前发过一篇MHA的文章，介绍了MHA相关的知识和功能测试，连接为：【DB宝19】在Docker中使用MySQL高可用之MHA 。今天这一篇给大家分享一下“MHA+中间件ProxySQL”来实现读写分离+负载均衡的相关知识。

 AiDBA宝典 · 2021/03/09 522 0

ProxySQL简介原理及读写分离应用

云数据库 SQL Server

https://www.cnblogs.com/you-men/p/12838333.html



iginkgo18

· 2020/09/27

2.3K

0



MYSQL PROXYSQL "集群"怎么搞 与 灵活的proxysql 应用

sql server 网络安全 sql 数据库 云数据库 SQL Server

PROXYSQL 是一个集中化的服务节点,在部署PROXYSQL的时候大部分都是单节点来部署,或者和应用节点一起部署.

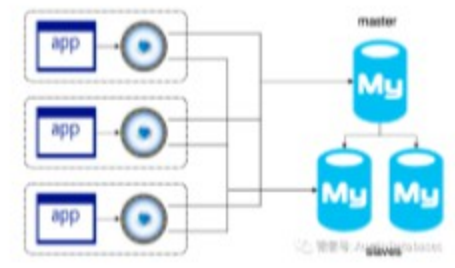


AustinDatabases

· 2020/10/30

1.9K

0



【DB宝42】MySQL高可用架构MHA+ProxySQL实现读写分离和负载均衡（上）

云数据库 SQL Server 负载均衡 负载均衡缓存 容器镜像服务

之前发过一篇MHA的文章，介绍了MHA相关的知识和功能测试，连接为：【DB宝19】在Docker中使用MySQL高可用之MHA 。今天这一篇给大家分享一下“MHA+中间件...



AIDBA宝典

· 2021/03/08

686

0



proxy SQL实现MySQL主从读写分离

网络安全 sql 数据库 云数据库 SQL Server

ProxySQL 是一款可以实际用于生产环境的 MySQL 中间件，它有官方版和 percona版两种。percona版是在官方版的基础上修改的，添加了几个比较实用的工具。本文...

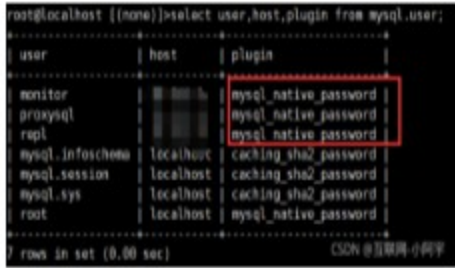


互联网-小阿宇

· 2022/11/21

1.8K

0



Proxy实现mysql读写分离

sqlite sql server android 网络安全 sql

ProxySQL 是一款可以实际用于生产环境的 MySQL 中间件，它有官方版和 percona版两种。percona版是在官方版的基础上修改的，添加了几个比较实用的工具。生产...

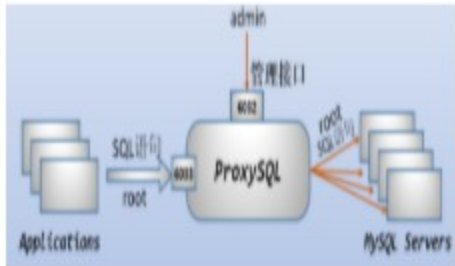


全栈程序员站长

· 2022/07/25

2.3K

0



ProxySQL！像C罗一样的强大！

sql 数据库 云数据库 SQL Server mongodb

作者 | 张甦， 数据库领域的专家和知名人士、图书《MySQL王者晋级之路》作者，51CTO 专家博主。近10年互联网线上处理及培训经验，专注于 MySQL 数据库，对...



数据和云

· 2018/07/27

1.4K

0



图文结合 | GreatSQL MGR + ProxySQL集群搭建方案

📁 数据库 集群 监控 路由 配置

ProxySQL 是基于 MySQL 的一款开源的中间件的产品，是一个灵活的 MySQL 代理层，可以实现读写分离，支持 Query 路由功能，支持动态指定某个 SQL 进行缓存...

GreatSQL社区 · 2023/09/20

👁 380

💬 0



MYSQL Proxysql 2.X 添加了什么新功能

📁 网络安全 sql 数据库 云数据库 SQL Server 数据分析

从PROXYSQL诞生起，发展有了些年头了，目前已经发展到了2.X，从以前一个人，到现在成立了公司，更加完善和强大了PROXYSQL 已经成为很多人的选择。

AustinDatabases · 2020/10/30

👁 818

💬 0



社区

- 技术文章
- 技术问答
- 技术沙龙
- 技术视频
- 学习中心
- 技术百科
- 技术专区

活动

- 自媒体同步曝光计划
- 邀请作者入驻
- 自荐上首页
- 技术竞赛

圈层

- 腾讯云最具价值专家
- 腾讯云架构师技术同盟
- 腾讯云创作之星
- 腾讯云TDP

关于

- 社区规范
- 免责声明
- 联系我们
- 友情链接
- MCP广场开源版权声明

腾讯云开发者



扫码关注腾讯云开发者
领取腾讯云代金券

热门产品

- 域名注册
- 云数据库
- 云服务器
- 域名解析
- 区块链服务
- 云存储
- 消息队列
- 视频直播
- 网络加速

热门推荐

- 人脸识别
- 图像分析
- 腾讯会议
- MySQL 数据库
- 企业云
- SSL 证书
- CDN加速
- 语音识别
- 视频通话

更多推荐

- 数据安全
- 大数据
- 负载均衡
- 小程序开发
- 短信
- 网站监控
- 文字识别
- 数据迁移
- 云点播

Copyright © 2013 – 2025 Tencent Cloud. All Rights Reserved. 腾讯云 版权所有

深圳市腾讯计算机系统有限公司 ICP备案/许可证号：粤B2–20090059 深公网安备号 44030502008569

腾讯云计算（北京）有限责任公司 京ICP证150476号 | 京ICP备11018762号 | 京公网安备号11010802020287