# Mysql process

这种只能看同一个账号下面的process

mysql>show processlist;

出来哗啦啦好几屏幕的, 没有一千也有几百条, 查询语句把表锁住了, 赶紧找出第一个Locked的thread\_id, 在mysql的shell里面执行.

mysql>kill thread\_id;

kill掉第一个锁表的进程, 依然没有改善. 既然不改善, 咱们就想办法将所有锁表的进程kill掉吧, 简单的脚本如下.

#!/bin/bash

mysql -u root -e "show processlist" | grep -i "Locked" >> locked\_log.txt

for line in `cat locked\_log.txt | awk '{print $1}'`

do

echo "kill $line;" >> kill\_thread\_id.sql

done

现在kill\_thread\_id.sql的内容像这个样子

kill 66402982;

kill 66402983;

kill 66402986;

kill 66402991;

.....

好了, 我们在mysql的shell中执行, 就可以把所有锁表的进程杀死了.

mysql>source kill\_thread\_id.sql

当然了, 也可以一行搞定

for id in `mysqladmin processlist | grep -i locked | awk '{print $1}'`

do

mysqladmin kill ${id}

done

1、Sleep

通常代表资源未释放，如果是通过连接池，sleep状态应该恒定在一定数量范围内

实战范例：因前端数据输出时（特别是输出到用户终端）未及时关闭数据库连接，导致因网络连接速度产生大量sleep连接，在网速出现异常时，数据库too many connections挂死。

简单解读，数据查询和执行通常只需要不到0.01秒，而网络输出通常需要1秒左右甚至更长，原本数据连接在0.01秒即可释放，但是因为前端程序未执行close操作，直接输出结果，那么在结果未展现在用户桌面前，该数据库连接一直维持在sleep状态！

2、Waiting for net, reading from net, writing to net

偶尔出现无妨, 如大量出现，迅速检查数据库到前端的网络连接状态和流量

案例:因外挂程序，内网数据库大量读取，内网使用的百兆交换迅速爆满，导致大量连接阻塞在waiting for net，数据库连接过多崩溃

3、Locked

有更新操作锁定

通常使用innodb可以很好的减少locked状态的产生，但是切记，更新操作要正确使用索引，即便是低频次更新操作也不能疏忽。

在myisam的时代，locked是很多高并发应用的噩梦。所以mysql官方也开始倾向于推荐innodb。

4、Copy to tmp table

索引及现有结构无法涵盖查询条件，才会建立一个临时表来满足查询要求，产生巨大的恐怖的i/o压力。

很可怕的搜索语句会导致这样的情况，如果是数据分析，或者半夜的周期数据清理任务，偶尔出现，可以允许。频繁出现务必优化之。

Copy to tmp table通常与连表查询有关，建议逐渐习惯不使用连表查询。

范例：

某社区数据库阻塞，求救，经查，其服务器存在多个数据库应用和网站，其中一个不常用的小网站数据库产生了一个恐怖的copy to tmp table操作，导致整个硬盘i/o和cpu压力超载。Kill掉该操作一切恢复。

5、Sending data

Sending data并不是发送数据，别被这个名字所欺骗，这是从物理磁盘获取数据的进程，如果你的影响结果集较多，那么就需要从不同的磁盘碎片去抽取数据，

偶尔出现该状态连接无碍。

回到上面影响结果集的问题，一般而言，如果sending data连接过多，通常是某查询的影响结果集过大，也就是查询的索引项不够优化。

如果出现大量相似的SQL语句出现在show proesslist列表中，并且都处于sending data状态，优化查询索引，记住用影响结果集的思路去思考。

6、Storing result to query cache

出现这种状态，如果频繁出现，使用set profiling分析，如果存在资源开销在SQL整体开销的比例过大（即便是非常小的开销，看比例），则说明query cache碎片较多

使用flush query cache可即时清理，也可以做成定时任务

Query cache参数可适当酌情设置。

7、Freeing items

理论上这玩意不会出现很多。偶尔出现无碍

如果大量出现，内存，硬盘可能已经出现问题，比如硬盘满或损坏，或者执行正在进行清理 更新日志索引等，i/o压力过大时，也可能出现Free items执行时间较长的情况。

数据库出现大量的freeing items状态 表更新慢 而且大量锁表

8、Sorting for …

和Sending data类似，结果集过大，排序条件没有索引化，需要在内存里排序，甚至需要创建临时结构排序。

9、Closing tables

正在将表中修改的数据刷新到磁盘中，同时正在关闭已经用完的表。这是一个很快的操作，如果不是这样的话，就应该确认磁盘空间是否已经满了或者磁盘是否正处于重负中。

10、Connect Out

复制从服务器正在连接主服务器。

11、Killed

发送了一个kill请求给某线程，那么这个线程将会检查kill标志位，同时会放弃下一个kill请求。MySQL会在每次的主循环中检查kill标志位，不过有些情况下该线程可能会过一小段才能死掉。如果该线程程被其他线程锁住了，那么kill请求会在锁释放时马上生效。

12、Opening tables

这个过程应该会很快，除非受到其他因素的干扰。例如，在执ALTER TABLE或LOCK TABLE语句行完以前，数据表无法被其他线程打开。正尝试打开一个表。

13、Removing duplicates

　 正在执行一个SELECT DISTINCT方式的查询，但是MySQL无法在前一个阶段优化掉那些重复的记录。因此，MySQL需要再次去掉重复的记录，然后再把结果发送给客户端。

14、System lock

正在等待取得一个外部的系统锁。如果当前没有运行多个mysqld服务器同时请求同一个表，那么可以通过增加--skip-external-locking参数来禁止外部系统锁。

15、Waiting for tables

该线程得到通知，数据表结构已经被修改了，需要重新打开数据表以取得新的结构。然后，为了能的重新打开数据表，必须等到所有其他线程关闭这个表。

以下几种情况下会产生这个通知：FLUSH TABLES tbl\_name, ALTER TABLE, RENAME TABLE, REPAIR TABLE, ANALYZE TABLE,或OPTIMIZE TABLE。

————————————————

版权声明：本文为CSDN博主「plt305」的原创文章，遵循CC 4.0 BY-SA版权协议，转载请附上原文出处链接及本声明。

原文链接：https://blog.csdn.net/u013950690/article/details/129462036