

如何查询mysql中是否表被锁

2016年09月06日 09:41:10 [天府云创](#) 阅读数: 65177

版权声明: 本文为EnweiTech原创文章, 未经博主允许不得转载。

<https://blog.csdn.net/English0523/article/details/52447006>

可直接在mysql命令行执行: `show engine innodb status\G;`

查看造成死锁的sql语句, 分析索引情况, 然后优化sql然后show processlist;

`show status like '%lock%'`

`show OPEN TABLES where In_use > 0;` 这个语句记录当前锁表状态

另外可以打开慢查询日志, linux下打开需在my.cnf的[mysqld]里面加上以下内容:

`slow_query_log=TRUE`(有些mysql版本是ON)

`slow_query_log_file=/usr/local/mysql/slow_query_log.txt`

`long_query_time=3`

Windows:

在my.ini配置文件的[mysqld]选项下增加:

`slow_query_log=TRUE`

`slow_query_log_file=c:/slow_query_log.txt`

`long_query_time=3`

添加完成之后记得一定要重启mysql服务才能生效记录输出。

最后在MySQL客户端中输入命令:

`show variables like '%quer%';` 核查一哈是否Ok

[开启mysql慢查询日志](#)

查看配置:

//查看慢查询时间

`show variables like "long_query_time";` 默认10s

//查看慢查询配置情况

`show status like "%slow_queries%";`

//查看慢查询日志路径

`show variables like "%slow%";`

-slow_query_log是否记录慢查询。用long_query_time变量的值来确定“慢查询”。

-slow_query_log_file慢日志文件路径

-long_query_time慢日志执行时长（秒），超过设定的时间才会记日志

言归正传，回到我们问题的出发点上**Mysql 查询表死锁 和结束死锁的表步骤**

1、查询是否锁表 `show OPEN TABLES where In_use > 0;`

2、查询进程

`show processlist` 查询到相对应的进程===然后 `kill id`

补充：

查看正在锁的事务

`SELECT * FROM INFORMATION_SCHEMA.INNODB_LOCKS;`

查看等待锁的事务

`SELECT * FROM INFORMATION_SCHEMA.INNODB_LOCK_WAITS;`

【附录知识】

MySQL锁定状态查看命令

1 `show processlist;`

`SHOW PROCESSLIST`显示哪些线程正在运行。您也可以使用mysqladmin

`processlist`语句得到此信息。如果您有SUPER权限，您可以看到所有线程。否则，您只能看到您自己的线程（也就是，与您正在使用的MySQL账户相关的线程）。如果有线程在update或者insert 某个表，此时进程的状态为updating 或者 sending data。

如果您得到“too many connections” 错误信息，并且想要了解正在发生的情况，本语句是非常有用的。MySQL保留一个额外的连接，让拥有SUPER权限的账户使用，以确保管理员能够随时连接和检查系统（假设您没有把此权限给予所有的用户）。

Status	含义

Checking table	正在检查数据表（这是自动的）。
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。确认磁盘空间是否已经满了或者磁盘是否正处于重负中。
Connect Out	复制从服务器正在连接主服务器。
Copying to tmp table on disk	由于临时结果集大于tmp_table_size，正在将临时表从内存存储转为磁盘存储。
Creating tmp table	正在创建临时表以存放部分查询结果。
deleting from main table	服务器正在执行多表删除中的第一部分，刚删除第一个表。
deleting from reference tables	服务器正在执行多表删除中的第二部分，正在删除其他表的记录。
Flushing tables	正在执行FLUSH TABLES，等待其他线程关闭数据表。
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亡，会在锁释放时马上生效。
Locked	被其他查询锁住了。
Sending data	正在处理SELECT查询的记录，同时正在把结果发送给客户端。
Sorting for group	正在为GROUP BY做排序。
Sorting for order	正在为ORDER BY做排序。
Opening tables	这个过程应该会很快，除非受到其他因素的干扰。例如，在执行ALTER TABLE被其他线程打开。正尝试打开一个表。
Removing duplicates	正在执行一个SELECT DISTINCT方式的查询，但是MySQL无法在前要再次去掉重复的记录，然后再把结果发送给客户端。
Reopen table	获得了对一个表的锁，但是必须在表结构修改之后才能获得这个锁。表。
Repair by sorting	修复指令正在排序以创建索引。
Repair with keycache	修复指令正在利用索引缓存一个一个地创建新索引。它会比Repair by sorting快。
Searching rows for update	正在讲符合条件的记录找出来以备更新。它必须在UPDATE要修改相关记录之前先取得锁。
Sleeping	正在等待客户端发送新请求。
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，则不需要取得这个锁。可以通过--external-locking参数来禁止外部系统锁。
Upgrading lock	INSERT DELAYED正在尝试取得一个锁表以插入新记录。
Updating	正在搜索匹配的记录，并且修改它们。
User Lock	正在等待GET_LOCK()。

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 RENAME TABLE, REPAIR TABLE, ANALYZE TABLE,或OPTIMIZE TABLE
waiting for handler insert	INSERT DELAYED已经处理完了所有待处理的插入操作，正在等待

大部分状态对应很快的操作，只要有一个线程保持同一个状态好几秒钟，那么可能是有问题发生了，需要检查一下。还有其他的状态没在上面中列出来，不过它们大部分只是在查看服务器是否有存在错误是才用得着。

2 show full processlist;

show processlist;只列出前100条，如果想全列出请使用show full processlist;

3 show open tables;

这条命令能够查看当前有那些表是打开的。In_use列表示有多少线程正在使用某张表，Name_locked表示表名是否被锁，这一般发生在Drop或Rename命令操作这张表时。所以这条命令不能帮助解答我们常见的问题：当前某张表是否有死锁，谁拥有表上的这个锁等。

show open tables from database;

```
mysql> show open tables from test;
+-----+-----+-----+-----+
| Database | Table           | In_use | Name_locked |
+-----+-----+-----+-----+
| test     | stock_sales     | 0      | 0           |
| test     | stock_sales_book | 0      | 0           |
| test     | person          | 0      | 0           |
| test     | product_info    | 0      | 0           |
+-----+-----+-----+-----+
4 rows in set (0.04 sec)
```

4 show status like '%lock%'

查看服务器状态。

```
mysql> show status like '%lock%';
```

Variable_name	Value
Com_lock_tables	0
Com_unlock_tables	0
Innodb_row_lock_current_waits	0
Innodb_row_lock_time	357727
Innodb_row_lock_time_avg	51103
Innodb_row_lock_time_max	51418
Innodb_row_lock_waits	7
Key_blocks_not_flushed	0
Key_blocks_unused	214342
Key_blocks_used	0
Qcache_free_blocks	20
Qcache_total_blocks	268
Table_locks_immediate	18347
Table_locks_waited	0

```
14 rows in set (0.04 sec)
```

5 show engine innodb status\G;

MySQL 5.1之前的命令是：show innodbstatus\G;，MySQL 5.5使用上面命令即可查看innodb引擎的运行时信息。

```

mysql> show engine innodb status\G;
***** 1. row *****
Type: InnoDB
Name:
Status:
=====
130206 13:51:46 INNODB MONITOR OUTPUT
=====
Per second averages calculated from the last 13 seconds
-----
BACKGROUND THREAD
-----
srv_master_thread loops: 1 1_second, 1 sleeps, 0 10_second, 1 background, 1 flush
srv_master_thread log flush and writes: 1
-----
SEMAPHORES
-----
OS WAIT ARRAY INFO: reservation count 2, signal count 2
Mutex spin waits 0, rounds 0, OS waits 0
RW-shared spins 2, rounds 60, OS waits 2
RW-excl spins 0, rounds 0, OS waits 0
Spin rounds per wait: 0.00 mutex, 30.00 RW-shared, 0.00 RW-excl
-----
TRANSACTIONS
-----
Trx id counter 5300
Purge done for trx's n:o < 4F1B undo n:o < 0
History list length 215
LIST OF TRANSACTIONS FOR EACH SESSION:
---TRANSACTION 0, not started
MySQL thread id 1, OS thread handle 0x19a8, query id 19 localhost 127.0.0.1 root
show engine innodb status
-----
FILE I/O
-----
I/O thread 0 state: wait Windows aio (insert buffer thread)
I/O thread 1 state: wait Windows aio (log thread)
I/O thread 2 state: wait Windows aio (read thread)
I/O thread 3 state: wait Windows aio (read thread)
I/O thread 4 state: wait Windows aio (read thread)
I/O thread 5 state: wait Windows aio (read thread)
I/O thread 6 state: wait Windows aio (write thread)
I/O thread 7 state: wait Windows aio (write thread)
I/O thread 8 state: wait Windows aio (write thread)
I/O thread 9 state: wait Windows aio (write thread)
Pending normal aio reads: 0 [0, 0, 0, 0] , aio writes: 0 [0, 0, 0, 0] ,
ibuf aio reads: 0, log i/o's: 0, sync i/o's: 0
Pending flushes (fsync) log: 0; buffer pool: 0
349 OS file reads, 3 OS file writes, 3 OS fsyncs
0.00 reads/s, 0 avg bytes/read, 0.00 writes/s, 0.00 fsyncs/s
-----
INSERT BUFFER AND ADAPTIVE HASH INDEX
-----
Ibuf: size 1, free list len 0, seg size 2, 0 merges
merged operations:
insert 0, delete mark 0, delete 0
discarded operations:
insert 0, delete mark 0, delete 0

```

6 show variables like ‘%timeout%’ ;

查看服务器配置参数。

```
mysql> show variables like '%timeout%';
+-----+-----+
| Variable_name | Value |
+-----+-----+
| connect_timeout | 10 |
| delayed_insert_timeout | 300 |
| innodb_lock_wait_timeout | 50 |
| innodb_rollback_on_timeout | OFF |
| interactive_timeout | 7200 |
| net_read_timeout | 30 |
| net_write_timeout | 60 |
| slave_net_timeout | 3600 |
| table_lock_wait_timeout | 50 |
| wait_timeout | 7200 |
+-----+-----+
10 rows in set (0.04 sec)
```

【参考资料】

1. 慢查询日志分析 - wulantian的专栏 - 博客频道 -

CSDN.NET <http://blog.csdn.net/wulantian/article/details/40296541>

2.mysql5.0经常出现 err=1205 - Lockwait timeout exceeded; try restarting transaction

<http://www.imysql.cn/node/165>

3.mysql show processlist命令详解

<http://www.cnblogs.com/JulyZhang/archive/2011/01/28/1947165.html>

4.MySQL锁

http://blog.csdn.net/c_ilikeyouma/article/details/8541195

5.SHOW INNODB STATUS提示语法错误?

<http://www.itpub.net/thread-1454597-1-1.html>

6.SHOW OPEN TABLES – what is in your tablecache

http://blog.sina.com.cn/s/blog_4d1f40c00100rsse.html