mysql锁调试

锁等待可能发生在服务器级别或者存储引擎级别。

# 服务器级别的锁调试

表锁：表可以被显式的读锁和写锁进行锁定。查询过程还有隐式的锁。

全局锁：可以通过flush tables with read lock或设置read\_only=1来获取全局读锁，它与任何表锁都冲突。

命名锁：表锁的一种，服务器在重命名或者删除一个表时创建。

字符锁：可以使用get\_lock()及其相关函数在服务器级别内锁住和释放任意一个字符串。

## 表锁

* 显示表锁

|  |
| --- |
| #mysql只能查询锁定表中的数据，没在锁定表之内的访问都会被拒绝  #read和write的区别在于允许在锁定表中读或者读写的权限。  lock tables [table1] read,[table2] write,[table3] read;  #解除表锁  unlock tables或者kill线程 |

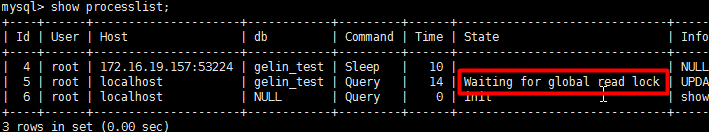
* 隐式表锁

服务器在查询过程中会隐式的锁住表。从mysql内部结构中，隐式锁和显示锁有着相同结构，由相同的MySQL服务器代码来控制。

## 全局读锁

#全局读锁，库中所有的表只有读取的权限

**FLUSH** **TABLES** **WITH** **READ** **LOCK**;



#解锁

unlock tables

* 命名锁

命名锁是表锁的一种：服务器在重命名或者删除一个表的时候创建。

# InnoDB中的锁等待

* 显示谁阻塞和谁等待，以及等待多久

|  |
| --- |
| **SELECT**  r.trx\_id waiting\_trx\_id,  r.trx\_mysql\_thread\_id waiting\_thread, **TIMESTAMPDIFF**(**SECOND**,  r.trx\_wait\_started, **CURRENT\_TIMESTAMP**  ) wait\_time,  r.trx\_query waiting\_query,  l.lock\_table waiting\_table\_lock,  b.trx\_id blocking\_trx\_id,  b.trx\_mysql\_thread\_id blocking\_thread, **SUBSTRING**(  p. HOST,  1, **INSTR**(p. HOST, ':') - 1  ) blocking\_host, **SUBSTRING**(p. HOST, **INSTR**(p. HOST, ':') + 1) blocking\_port, **IF** (p.COMMAND = 'Sleep', p.**TIME**, 0) idel\_in\_trx,  b.trx\_query blocking\_query  **FROM**  information\_schema.INNODB\_LOCK\_WAITS w  **INNER** **JOIN** information\_schema.INNODB\_TRX b **ON** b.trx\_id = w.blocking\_trx\_id  **INNER** **JOIN** information\_schema.INNODB\_TRX r **ON** r.trx\_id = w.requesting\_trx\_id  **INNER** **JOIN** information\_schema.INNODB\_LOCKS l **ON** w.requested\_lock\_id = l.lock\_id  **LEFT** **JOIN** information\_schema. **PROCESSLIST** p **ON** p.ID = b.trx\_mysql\_thread\_id  **ORDER** **BY**  wait\_time **DESC**; |

结果显示：线程(waiting\_thread)已经等待查询(waiting\_query)达到(wait\_time)s.它在线程(blocking\_thread)上阻塞，而且该线程被空闲(idel\_in\_trx)s

* 多少查询被哪些线程阻塞

|  |
| --- |
| **SELECT** **CONCAT**('thread ', b.trx\_mysql\_thread\_id, ' from ', p.host) **AS** who\_blocks, **IF**(p.command = "Sleep", p.**time**, 0) **AS** idle\_in\_trx, **MAX**(**TIMESTAMPDIFF**(**SECOND**, r.trx\_wait\_started, **NOW**())) **AS** max\_wait\_time, **COUNT**(\*) **AS** num\_waiters  **FROM** INFORMATION\_SCHEMA.INNODB\_LOCK\_WAITS **AS** w  **INNER** **JOIN** INFORMATION\_SCHEMA.INNODB\_TRX **AS** b **ON** b.trx\_id = w.blocking\_trx\_id  **INNER** **JOIN** INFORMATION\_SCHEMA.INNODB\_TRX **AS** r **ON** r.trx\_id = w.requesting\_trx\_id  **LEFT** **JOIN** INFORMATION\_SCHEMA.**PROCESSLIST** **AS** p **ON** p.id = b.trx\_mysql\_thread\_id  **GROUP** **BY** who\_blocks  **ORDER** **BY** num\_waiters **DESC** |

结果：显示线程(who\_blocks)已经空闲了更长一段时间，并且至少有1个线程已经等待它释放的锁长达(max\_wait\_time)s。有(num\_waiters)个线程在等待线程(who\_blocks)完成它的工作并提交。