# slow log的分析

## slow log的内容例子

Time Id Command Argument

# Time: 141202 14:17:36

# User@Host: root[root] @ localhost [] Id: 1

# Query\_time: 2.486444 Lock\_time: 0.000912 Rows\_sent: 1 Rows\_examined: 0

SET timestamp=1417501056;

select count(\*) from performance\_schema.events\_statements\_history;

## slow log的重要时间点

语句开始执行的时间点（1）

获得表lock的时间点（2），现已知有两处

判断是否为slow query的时间点（3）

计算Query\_time和Lock\_time的时间点（4）

按照时间点可以分为两个时间段，开始执行到获取表锁的时间段（如果有其他事务长期占用该表，则此时间段也变长，两者时间成正比）和语句执行的时间段（由于其中3和4的时间点在语句执行完立刻结束，所以）

时间点3是这样判断是否为slow query的

ulonglong end\_utime\_of\_query= current\_utime();

if (end\_utime\_of\_query > utime\_after\_lock + variables.long\_query\_time)

server\_status|= SERVER\_QUERY\_WAS\_SLOW;

utime\_after\_lock为时间点2记录的时间，variables.long\_query\_time为配置文件（命令行指定的long\_query\_time的内容），end\_utime\_of\_query为当前时间

时间点4是这样生成Query\_time和Lock\_time的

current\_utime= thd->current\_utime();

current\_time= my\_time\_possible\_from\_micro(current\_utime);

if (thd->start\_utime)

{

query\_utime= (current\_utime - thd->start\_utime);

lock\_utime= (thd->utime\_after\_lock - thd->start\_utime);

}

query\_utime为打印的Query\_time，lock\_utime为打印的Lock\_time

结论是：

打印出的Query\_time为执行的总时间（包括等待锁的时间），Lock\_time为等待锁的时间；判断是否为slow query是获取所需的表锁之后，进行执行的时间（如果是多表则是最后一个获取到的表锁的时间）

### 变量

start\_utime，保存语句开始执行的时间点

utime\_after\_lock，保存获得表lock的时间点

### 时间点对应的函数

THD::set\_time，开始记录时间的函数，对应

THD::update\_server\_status：用于进行是否slow log的更新

THD::update\_server\_status

LOGGER::slow\_log\_print

### 时间点1的堆栈电梯

#0 THD::set\_time (this=0x1d67bc0)

at /home/jiangyx/mysql/ops/mysql/mysql-5-6-12-ctrip/sql/sql\_class.h:3220

#1 0x00000000007a5163 in dispatch\_command (command=COM\_QUERY, thd=0x1d67bc0,

packet=0x1e39e31 "select count(\*) from performance\_schema.events\_statements\_history", packet\_length=65)

at /home/jiangyx/mysql/ops/mysql/mysql-5-6-12-ctrip/sql/sql\_parse.cc:1163

#2 0x00000000007a4ddf in do\_command (thd=0x1d67bc0)

at /home/jiangyx/mysql/ops/mysql/mysql-5-6-12-ctrip/sql/sql\_parse.cc:1036

#3 0x000000000076f4a6 in do\_handle\_one\_connection (thd\_arg=0x1d67bc0)

at /home/jiangyx/mysql/ops/mysql/mysql-5-6-12-ctrip/sql/sql\_connect.cc:977

#4 0x000000000076efa3 in handle\_one\_connection (arg=0x1d67bc0)

at /home/jiangyx/mysql/ops/mysql/mysql-5-6-12-ctrip/sql/sql\_connect.cc:893

#5 0x0000000000aaa9d4 in pfs\_spawn\_thread (arg=0x1de5820)

at /home/jiangyx/mysql/ops/mysql/mysql-5-6-12-ctrip/storage/perfschema/pfs.cc:1855

### 时间点2的堆栈电梯

#### 时间点2的第一个计时处堆栈

#0 THD::set\_time\_after\_lock (this=0x1d69da0)

at /home/jiangyx/mysql/ops/mysql/mysql-5-6-12-ctrip/sql/sql\_class.h:3255

#1 0x0000000000907946 in mysql\_lock\_tables (thd=0x1d69da0,

tables=0x7fffc4005e78, count=1, flags=0)

at /home/jiangyx/mysql/ops/mysql/mysql-5-6-12-ctrip/sql/lock.cc:341

#2 0x0000000000745d1f in lock\_tables (thd=0x1d69da0, tables=0x7fffc4005348,

count=1, flags=0)

at /home/jiangyx/mysql/ops/mysql/mysql-5-6-12-ctrip/sql/sql\_base.cc:5831

#3 0x00000000007d5a07 in mysql\_select (thd=0x1d69da0, tables=0x7fffc4005348,

wild\_num=0, fields=..., conds=0x0, order=0x1d6c760, group=0x1d6c698,

having=0x0, select\_options=2147748608, result=0x7fffc4005958,

unit=0x1d6bf50, select\_lex=0x1d6c598)

at /home/jiangyx/mysql/ops/mysql/mysql-5-6-12-ctrip/sql/sql\_select.cc:1201

#4 0x00000000007d3b6f in handle\_select (thd=0x1d69da0, result=0x7fffc4005958,

setup\_tables\_done\_option=0)

at /home/jiangyx/mysql/ops/mysql/mysql-5-6-12-ctrip/sql/sql\_select.cc:110

#5 0x00000000007afe36 in execute\_sqlcom\_select (thd=0x1d69da0,

all\_tables=0x7fffc4005348)

at /home/jiangyx/mysql/ops/mysql/mysql-5-6-12-ctrip/sql/sql\_parse.cc:5046

#6 0x00000000007a89f1 in mysql\_execute\_command (thd=0x1d69da0)

at /home/jiangyx/mysql/ops/mysql/mysql-5-6-12-ctrip/sql/sql\_parse.cc:2604

#7 0x00000000007b2369 in mysql\_parse (thd=0x1d69da0,

rawbuf=0x7fffc4004fd0 "select count(\*) from performance\_schema.events\_statements\_history", length=65, parser\_state=0x7ffff40a1660)

at /home/jiangyx/mysql/ops/mysql/mysql-5-6-12-ctrip/sql/sql\_parse.cc:6187

#8 0x00000000007a5c30 in dispatch\_command (command=COM\_QUERY, thd=0x1d69da0,

packet=0x1e3c2b1 "select count(\*) from performance\_schema.events\_statements\_history", packet\_length=65)

at /home/jiangyx/mysql/ops/mysql/mysql-5-6-12-ctrip/sql/sql\_parse.cc:1334

#9 0x00000000007a4ddf in do\_command (thd=0x1d69da0)

at /home/jiangyx/mysql/ops/mysql/mysql-5-6-12-ctrip/sql/sql\_parse.cc:1036

#10 0x000000000076f4a6 in do\_handle\_one\_connection (thd\_arg=0x1d69da0)

at /home/jiangyx/mysql/ops/mysql/mysql-5-6-12-ctrip/sql/sql\_connect.cc:977

#11 0x000000000076efa3 in handle\_one\_connection (arg=0x1d69da0)

at /home/jiangyx/mysql/ops/mysql/mysql-5-6-12-ctrip/sql/sql\_connect.cc:893

#12 0x0000000000aaa9d4 in pfs\_spawn\_thread (arg=0x1e235a0)

at /home/jiangyx/mysql/ops/mysql/mysql-5-6-12-ctrip/storage/perfschema/pfs.cc:1855

#### 时间点2的第二个计时处堆栈

发生在innodb有其他事务锁表的情况

#0 thd\_storage\_lock\_wait (thd=0x1e40610, value=25434421)

at /home/jiangyx/mysql/ops/mysql/mysql-5-6-12-ctrip/sql/sql\_class.cc:659

#1 0x0000000000ac1066 in thd\_set\_lock\_wait\_time (thd=0x1e40610,

value=25434421)

at /home/jiangyx/mysql/ops/mysql/mysql-5-6-12-ctrip/storage/innobase/handler/ha\_innodb.cc:1263

#2 0x0000000000b20ca9 in lock\_wait\_suspend\_thread (thr=0x7fffc4033a58)

at /home/jiangyx/mysql/ops/mysql/mysql-5-6-12-ctrip/storage/innobase/lock/lock0wait.cc:372

#3 0x0000000000bad443 in row\_mysql\_handle\_errors (new\_err=0x7ffff405e964,

trx=0x7fffc8007d78, thr=0x7fffc4033a58, savept=0x0)

at /home/jiangyx/mysql/ops/mysql/mysql-5-6-12-ctrip/storage/innobase/row/row0mysql.cc:630

#4 0x0000000000bdb9e6 in row\_search\_for\_mysql (buf=0x7fffc402f6e0 "\377\n",

mode=1, prebuilt=0x7fffc4033068, match\_mode=0, direction=1)

at /home/jiangyx/mysql/ops/mysql/mysql-5-6-12-ctrip/storage/innobase/row/row0sel.cc:5017

#5 0x0000000000acb097 in ha\_innobase::general\_fetch (this=0x7fffc402dbc0,

buf=0x7fffc402f6e0 "\377\n", direction=1, match\_mode=0)

at /home/jiangyx/mysql/ops/mysql/mysql-5-6-12-ctrip/storage/innobase/handler/ha\_innodb.cc:7705

#6 0x0000000000acb2b5 in ha\_innobase::index\_next (this=0x7fffc402dbc0,

buf=0x7fffc402f6e0 "\377\n")

at /home/jiangyx/mysql/ops/mysql/mysql-5-6-12-ctrip/storage/innobase/handler/ha\_innodb.cc:7767

#7 0x000000000061e053 in handler::ha\_index\_next (this=0x7fffc402dbc0,

buf=0x7fffc402f6e0 "\377\n")

at /home/jiangyx/mysql/ops/mysql/mysql-5-6-12-ctrip/sql/handler.cc:2751

#8 0x000000000077e9fd in join\_read\_next (info=0x7fffc80065a8)

at /home/jiangyx/mysql/ops/mysql/mysql-5-6-12-ctrip/sql/sql\_executor.cc:2542

#9 0x000000000077bb51 in sub\_select (join=0x7fffc8005460,

join\_tab=0x7fffc8006518, end\_of\_records=false)

at /home/jiangyx/mysql/ops/mysql/mysql-5-6-12-ctrip/sql/sql\_executor.cc:1255

#10 0x000000000077b579 in do\_select (join=0x7fffc8005460)

at /home/jiangyx/mysql/ops/mysql/mysql-5-6-12-ctrip/sql/sql\_executor.cc:930

#11 0x00000000007794f1 in JOIN::exec (this=0x7fffc8005460)

at /home/jiangyx/mysql/ops/mysql/mysql-5-6-12-ctrip/sql/sql\_executor.cc:191

#12 0x00000000007d57b4 in mysql\_execute\_select (thd=0x1e40610,

select\_lex=0x1e42e08, free\_join=true)

at /home/jiangyx/mysql/ops/mysql/mysql-5-6-12-ctrip/sql/sql\_select.cc:1100

#13 0x00000000007d5aa6 in mysql\_select (thd=0x1e40610, tables=0x7fffc8004e28,

wild\_num=1, fields=..., conds=0x0, order=0x1e42fd0, group=0x1e42f08,

having=0x0, select\_options=2147748608, result=0x7fffc8005438,

unit=0x1e427c0, select\_lex=0x1e42e08)

at /home/jiangyx/mysql/ops/mysql/mysql-5-6-12-ctrip/sql/sql\_select.cc:1221

#14 0x00000000007d3b6f in handle\_select (thd=0x1e40610, result=0x7fffc8005438,

setup\_tables\_done\_option=0)

at /home/jiangyx/mysql/ops/mysql/mysql-5-6-12-ctrip/sql/sql\_select.cc:110

#15 0x00000000007afe36 in execute\_sqlcom\_select (thd=0x1e40610,

all\_tables=0x7fffc8004e28)

at /home/jiangyx/mysql/ops/mysql/mysql-5-6-12-ctrip/sql/sql\_parse.cc:5046

#16 0x00000000007a89f1 in mysql\_execute\_command (thd=0x1e40610)

at /home/jiangyx/mysql/ops/mysql/mysql-5-6-12-ctrip/sql/sql\_parse.cc:2604

#17 0x00000000007b2369 in mysql\_parse (thd=0x1e40610,

rawbuf=0x7fffc8004c10 "select \* from trpldb.trpl1 for update", length=37,

parser\_state=0x7ffff4060660)

at /home/jiangyx/mysql/ops/mysql/mysql-5-6-12-ctrip/sql/sql\_parse.cc:6187

#18 0x00000000007a5c30 in dispatch\_command (command=COM\_QUERY, thd=0x1e40610,

packet=0x1e44281 "", packet\_length=37)

at /home/jiangyx/mysql/ops/mysql/mysql-5-6-12-ctrip/sql/sql\_parse.cc:1334

#19 0x00000000007a4ddf in do\_command (thd=0x1e40610)

at /home/jiangyx/mysql/ops/mysql/mysql-5-6-12-ctrip/sql/sql\_parse.cc:1036

#20 0x000000000076f4a6 in do\_handle\_one\_connection (thd\_arg=0x1e40610)

at /home/jiangyx/mysql/ops/mysql/mysql-5-6-12-ctrip/sql/sql\_connect.cc:977

#21 0x000000000076efa3 in handle\_one\_connection (arg=0x1e40610)

at /home/jiangyx/mysql/ops/mysql/mysql-5-6-12-ctrip/sql/sql\_connect.cc:893

#22 0x0000000000aaa9d4 in pfs\_spawn\_thread (arg=0x1d0bfe0)

at /home/jiangyx/mysql/ops/mysql/mysql-5-6-12-ctrip/storage/perfschema/pfs.cc:1855

### 时间点3的堆栈电梯

#0 THD::update\_server\_status (this=0x1d69f40)

at /home/jiangyx/mysql/ops/mysql/mysql-5-6-12-ctrip/sql/sql\_class.h:3268

#1 0x00000000007a6f26 in dispatch\_command (command=COM\_QUERY, thd=0x1d69f40,

packet=0x1e3c1f1 "", packet\_length=65)

at /home/jiangyx/mysql/ops/mysql/mysql-5-6-12-ctrip/sql/sql\_parse.cc:1727

#2 0x00000000007a4ddf in do\_command (thd=0x1d69f40)

at /home/jiangyx/mysql/ops/mysql/mysql-5-6-12-ctrip/sql/sql\_parse.cc:1036

#3 0x000000000076f4a6 in do\_handle\_one\_connection (thd\_arg=0x1d69f40)

at /home/jiangyx/mysql/ops/mysql/mysql-5-6-12-ctrip/sql/sql\_connect.cc:977

#4 0x000000000076efa3 in handle\_one\_connection (arg=0x1d69f40)

at /home/jiangyx/mysql/ops/mysql/mysql-5-6-12-ctrip/sql/sql\_connect.cc:893

#5 0x0000000000aaa9d4 in pfs\_spawn\_thread (arg=0x1e122e0)

at /home/jiangyx/mysql/ops/mysql/mysql-5-6-12-ctrip/storage/perfschema/pfs.cc:1855

### 时间点4的堆栈电梯

#0 LOGGER::slow\_log\_print (this=0x1770760, thd=0x1d67bc0,

query=0x7fffc4004fd0 "select count(\*) from performance\_schema.events\_statements\_history", query\_length=65)

at /home/jiangyx/mysql/ops/mysql/mysql-5-6-12-ctrip/sql/log.cc:1027

#1 0x00000000006ebc98 in slow\_log\_print (thd=0x1d67bc0,

query=0x7fffc4004fd0 "select count(\*) from performance\_schema.events\_statements\_history", query\_length=65)

at /home/jiangyx/mysql/ops/mysql/mysql-5-6-12-ctrip/sql/log.cc:2062

#2 0x00000000007a7388 in log\_slow\_do (thd=0x1d67bc0)

at /home/jiangyx/mysql/ops/mysql/mysql-5-6-12-ctrip/sql/sql\_parse.cc:1846

#3 0x00000000007a73e3 in log\_slow\_statement (thd=0x1d67bc0)

at /home/jiangyx/mysql/ops/mysql/mysql-5-6-12-ctrip/sql/sql\_parse.cc:1870

#4 0x00000000007a7040 in dispatch\_command (command=COM\_QUERY, thd=0x1d67bc0,

packet=0x1e39e31 "", packet\_length=65)

at /home/jiangyx/mysql/ops/mysql/mysql-5-6-12-ctrip/sql/sql\_parse.cc:1741

#5 0x00000000007a4ddf in do\_command (thd=0x1d67bc0)

at /home/jiangyx/mysql/ops/mysql/mysql-5-6-12-ctrip/sql/sql\_parse.cc:1036

#6 0x000000000076f4a6 in do\_handle\_one\_connection (thd\_arg=0x1d67bc0)

at /home/jiangyx/mysql/ops/mysql/mysql-5-6-12-ctrip/sql/sql\_connect.cc:977

#7 0x000000000076efa3 in handle\_one\_connection (arg=0x1d67bc0)

at /home/jiangyx/mysql/ops/mysql/mysql-5-6-12-ctrip/sql/sql\_connect.cc:893

#8 0x0000000000aaa9d4 in pfs\_spawn\_thread (arg=0x1de5820)

at /home/jiangyx/mysql/ops/mysql/mysql-5-6-12-ctrip/storage/perfschema/pfs.cc:1855

## 例子说明

Slow log的计时设置为0.1秒

例子1

执行语句：

select sleep(1);

slow log

# Time: 141203 9:30:45

# User@Host: root[root] @ localhost [] Id: 4

# Query\_time: 1.000884 Lock\_time: 0.000000 Rows\_sent: 1 Rows\_examined: 0

SET timestamp=1417570245;

select sleep(1);

sleep为执行中的睡眠时间，超过0.1秒，所以被记录。由于没有锁等待，所以Lock\_time为0，总的执行时间为1秒

例子2

使用两个事务A，B进行验证，验证操作如下

执行序列 事务A 事务B

1 Begin;

2 select 1 from trpldb.trpl1 limit 1 for update;

3 select sleep(1),1 from trpldb.trpl1 limit 1 for update;

4 等待10秒

5 commit;

Slow log

# Time: 141203 9:29:17

# User@Host: root[root] @ localhost [] Id: 3

# Query\_time: 12.826126 Lock\_time: 11.825148 Rows\_sent: 1 Rows\_examined: 1

SET timestamp=1417570157;

select sleep(1),1 from trpldb.trpl1 limit 1 for update;

可以看出sleep(1)的执行时间是Query\_time减去Lock\_time，Lock\_time是“11.825148”也就是事务A上等待时间（手动计时存在些许误差，所以不完全准确）

通过以上两个例子说明，Query\_time是总执行时间，Lock\_time是总的锁等待时间，两者之差是语句获取到全部资源后的执行时间（切记不包括网络发送时间，包括该时间的需要参考perf）