# 1 定时观察mysql运行状态

mysql运行状态数据很多，值获取我们关心的数据：取查询数量、线程连接数、线程运行数。

(1) 编写获取mysql运行状态脚本文件

脚本文件status.sh内容如下：

|  |
| --- |
| #!/bin/sh  echo 'collecting (Queries, Threads\_connected, Threads\_running) from mysql every second......'  echo 'press CTRL+C to exit.'  while true  do  mysqladmin -uroot ext | awk '/Queries/{q=$4} /Threads\_connected/{c=$4} /Threads\_running/{r=$4} END{printf "%d %d %d\n", q,c,r}' >> status.txt  sleep 1  done |

(2) 执行脚本文件

**sh status.sh**

获取的数据保存在当前目录下的status.txt文件中

(3) 结束后对文件status.txt进行加工，把加工后的数据保存到文件statusResult.txt

**awk '{q=$1-last;last=$1} {printf("%d %d %d\n", q, $2, $3)}' bench.txt >> result.txt**

(4) 用exel显示图表数据

用notepad文本编辑器打开result.txt，用制表分隔符'\t'替换原来数据之间的空格分隔符，然后直接复制数据到exel表格，插入图标就可以绘制出图形来

# 2 定时观察mysql进程状态

(1) 编写脚本文件

文件process.sh内容如下：

|  |
| --- |
| #!/bin/sh  echo 'Verify that the PROFILING is open in mysql? ';  echo "press any key to start"  read  echo 'Collecting mysql process status every millisecond......'  echo 'press CTRL+C to exit.'  while true  do  mysql -uroot -e 'show processlist \G' | grep State >> process.txt  usleep 100000  done |

(2) 在mysql命令行打开profile功能

**SET PROFILING =1;**

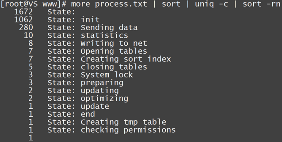
(3) 执行脚本

**sh process.sh**

每隔1毫秒获取一次进程状态，把状态数据写入process.txt

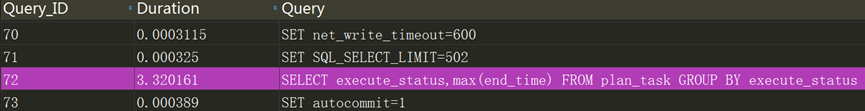
(4) 结束process.sh进程，对数据process.txt进行统计

**more process.txt | sort | uniq -c | sort -rn**



(5) 在mysql命令行查看哪些sql语句比较耗时，在phpstorm可以按时间顺序查看。

**SHOW PROFILES;**



附表:

|  |  |
| --- | --- |
| **状态名称** | **产生原因** |
| converting HEAP to MyISAM  查询结果太大时，把结果放在磁盘 | 一次性查询出来结果太大，内存装不下。例如select \* from big\_table; |
| create tmp table  创建临时表 | 一般合理利用索引就比较少出现创建临时表  (1) group by的列和order by的列不相同时，  (2) distinct和order by一起使用  (3) 开启了SQL\_SMALL\_RESULT选项 |
| copying to tmp table on disk  把内存临时表复制到磁盘 | (1) 取出的列含有text/blob类型时(内存表不支持)  (2) 在group by或distinct的列中存在>512字节的string类型的列  (3) select 中含有>512字节的string类型的列同时又使用了union或union all语句 |
| locked  被其他查询锁住 |  |
| logging slow query  记录慢查询 |  |
| sending data  发送数据 | 当查询结果比较多时，大部分时间都浪费在发送数据上，应该尽量减少 |