



链滴

mysql 基准测试脚本 (笔记)

作者: [spkinger](#)

原文链接: <https://ld246.com/article/1494518250533>

来源网站: 链滴

许可协议: [署名-相同方式共享 4.0 国际 \(CC BY-SA 4.0\)](#)

1.基准测试脚本

```
#!/bin/bash
INTERVAL=5
PREFIX=/home/imooc/benchmarks/$INTERVAL-sec-status
RUNFILE=/home/imooc/benchmarks/running
echo "1" > $RUNFILE
MYSQL=/usr/local/mysql/bin/mysql
$MYSQL -e "show global variables" >> mysql-variables
while test -e $RUNFILE; do
file=$(date +%F_%l)
sleep=$(date +%s.%N | awk '{print 5 - ($1 % 5)}')
sleep $sleep
ts=$(date +"TS %s.%N %F %T")
loadavg=$(uptime)
echo "$ts $loadavg" >> $PREFIX-{$file}-status
$MYSQL -e "show global status" >> $PREFIX-{$file}-status &
echo "$ts $loadavg" >> $PREFIX-{$file}-innodbstatus
$MYSQL -e "show engine innodb status" >> $PREFIX-{$file}-innodbstatus &
echo "$ts $loadavg" >> $PREFIX-{$file}-processlist
$MYSQL -e "show full processlist\G" >> $PREFIX-{$file}-processlist &
echo $ts
done
echo Exiting because $RUNFILE does not exists
```

2.测试结果分析脚本，此处获取QPS

```
#!/bin/bash
awk '
BEGIN {
printf "#ts date time load QPS";
fmt=" %.2f";
}
/^TS/ {
ts = substr($2,1,index($2,".")-1);
load = NF -2;
diff = ts - prev_ts;
printf "\n%s %s %s %s",ts,$3,$4,substr($load,1,length($load)-1);
prev_ts=ts;
}
/Queries/{
printf fmt,($2-Queries)/diff;
Queries=$2
}
' "$@"
```

3.mysql自带的测试工具：mysqlslap

```
--auto-generate-sql #由系统自动生成sql脚本测试
--auto-generate-sql-add-autoincrement #在生成的表中增加自增ID
--auto-generate-sql-load-type #指定测试中使用的查询类型
--auto-generate-sql-write-number #指定初始化时生成的数据量
--concurrency #指定并发线程的数量
--engine #指定存储引擎，可逗号分隔多个
```

```
--no-drop #指定不清理测试数据
--iterations #指定测试运行次数
--number-of-queries 指定每一个线程执行的查询数量
--debug-info #指定输出额外的内存及CPU统计信息
--number-int-cols #指定测试表中包含的int类型列的数量
--number-char-cols #指定测试表中包含的varchar类型列的数量
--create-schema #指定用于测试的数据库名称
--query #自定义SQL脚本
--only-print #只打印测试脚本，不测试
#例子
mysqlslap --concurrency=1,50,100,200 --iterations=3 --number-int-cols=5 --number-char-cols=5 --auto-generate-sql --auto-generate-sql-add-autoincrement --engine=myisam,innodb --number-of-queries=10 --create-schema=dbtest
```

4.测试工具：sysbench

安装

```
./autogen.sh
./configure --with-mysql-includes=/usr/local/mysql/include/ --with-mysql-libs=/usr/local/mysql/lib/
make && make install
```

常用参数

```
--test #指定索要执行的测试类型[Fileio：文件系统I/O性能测试，cpu：cpu性能测试，memory：内存性能测试，Oltpl：需要指定具体测试脚本，脚本位于 sysbench-0.5/sysbench/tests/db]
--mysql-db #用于指定执行基准测试的数据库名[要使用独立的测试数据库进行测试]
--mysql-table-engine #只能指定单个存储引擎
--oltp-tables-count #执行测试的表的数量
--oltp-tables-size #执行每个表中数据的行数
--num-threads #指定测试的并发线程数量
--max-time #测试执行最大时间就终止
--report-interval #指定多久输出一次统计信息，秒为单位
--mysql-user #指定测试的mysql用户
--mysql-password #测试用户的密码
```

```
prepare #准备测试数据
run #准备好后，执行的实际测试
cleanup #清理测试数据
```

例子：

```
sysbench --test=cpu --cpu-max-prime=10000 run
#测试系统I/O，要生成比系统内存大的文件
sysbench --test=fileio --file-total-size=1G prepare
sysbench --test=fileio --num-threads=8 --int-rng=on --file-total-size=1G --file-text-model=ndrw[随机读写] --report-interval=1 run
sysbench --test=./oltp.lua --mysql-table-engine=innodb --oltp-table-size=10000 --oltp-tables-count=10 --mysql-socket=/usr/local/mysql/data/mysql.sock --mysql-db=xxx --mysql-user xxx --mysql-password=xxx prepare
```

测试期间用1，2来分析结果