sysbench mysql和mariadb性能测试

sysbench是一个模块化的、跨平台、多线程基准测试工具，主要用于评估测试各种不同系统参数下的数据库负载情况。

主要测试方式					
cpu性能	磁盘io性能	调度程序性能	内存分配及传输速度	POSIX线程性能	数据库性能(OLTP基准测试)
找范围内最大素数{时间越短越好}	不同场景下IOPS{越大越好}	线程并发执行，循环响应信号量花费的时间{越少越好}	以不同块大小传输一定数量的数据吞吐量大小{越大越好}	并发线程同时申请互斥锁循环一定次数花费的时间{越少越好}	qps、tps越高越好

目前sysbench主要支持 MySQL,pgsql,oracle 这3种数据库。

安装测试环境为：RHEL7.2 MariaDB5.5.44

安装sysbench-0.5

源码地址https://github.com/BoobooWei/sysbench/archive/master.zip

```
[root@localhost ~]# unzip sysbench-master.zip
[root@localhost ~]# cd sysbench-master/
[root@localhost sysbench-master]# ls
autogen.sh  configure.ac  doc         Makefile.am    README.md         sysbench     TODO
ChangeLog   COPYING       install-sh  missing        README-Oracle.md  tests
config      debian        m4          mkinstalldirs  README-WIN.txt    third_party
[root@localhost sysbench-master]# yum install -y automake libtool

[root@localhost sysbench-master]# ./autogen.sh
./autogen.sh: running `libtoolize --copy --force'
libtoolize: putting auxiliary files in AC_CONFIG_AUX_DIR, `config'.
libtoolize: copying file `config/ltmain.sh'
libtoolize: putting macros in AC_CONFIG_MACRO_DIR, `m4'.
此处省略... ...
==========================================================================
sysbench version    : 1.0
CC                  : gcc
CFLAGS              : -O2 -ggdb3  -march=core2 -W -Wall -Wextra -Wpointer-arith -Wbad-function-
cast   -Wstrict-prototypes -Wnested-externs -Wno-inline -Wno-format-zero-length   -funroll-loops
-Wundef -Wstrict-prototypes -Wmissing-prototypes -Wmissing-declarations -Wredundant-decls -Wcast-
align        -pthread
CPPFLAGS            : -D_GNU_SOURCE   -I$(top_srcdir)/sysbench -
I$(abs_top_builddir)/third_party/luajit/inc -
I$(abs_top_builddir)/third_party/concurrency_kit/include
LDFLAGS             :
LIBS                : -lm
EXTRA_LDFLAGS       :

prefix              : /usr/local/sysbench
bindir              : ${prefix}/bin
libexecdir          : ${prefix}/libexec
mandir              : ${prefix}/share/man

MySQL support       : yes
Drizzle support     : no
AttachSQL support   : no
Oracle support      : no
PostgreSQL support  : no

LuaJIT              : bundled
LUAJIT_CFLAGS       : -I$(abs_top_builddir)/third_party/luajit/inc
LUAJIT_LIBS         : $(abs_top_builddir)/third_party/luajit/lib/libluajit-5.1.a -ldl
LUAJIT_LDFLAGS      : -rdynamic

Concurrency Kit     : bundled
CK_CFLAGS           : -I$(abs_top_builddir)/third_party/concurrency_kit/include
CK_LIBS             : $(abs_top_builddir)/third_party/concurrency_kit/lib/libck.a
configure flags     :
==========================================================================
[root@localhost sysbench-master]# make
[root@localhost sysbench-master]# make install
[root@localhost sysbench-master]# cd /usr/local/sysbench

[root@localhost sysbench]# pwd
```

```
/usr/local/sysbench
[root@localhost sysbench]# ls
bin  share
[root@localhost sysbench]# echo export PATH=$PATH:/usr/local/sysbench/bin >> /etc/bashrc
[root@localhost sysbench]# source /etc/bashrc
[root@localhost sysbench]# which sysbench
/usr/local/sysbench/bin/sysbench
# 测试的lua脚本存放位置
[root@localhost sysbench]# ls /root/sysbench-master/sysbench/tests/
CMakeLists.txt  db      Makefile     Makefile.in  mutex     sb_fileio.h  sb_mutex.h    threads
cpu             fileio  Makefile.am  memory       sb_cpu.h  sb_memory.h  sb_threads.h
[root@localhost sysbench]# ls /root/sysbench-master/sysbench/tests/db
bulk_insert.lua  insert.lua   Makefile.in      parallel_prepare.lua      select_random_ranges.lua
common.lua       Makefile     oltp.lua         select.lua                update_index.lua
delete.lua       Makefile.am  oltp_simple.lua  select_random_points.lua  update_non_index.lua
```

测试

MySQL数据库测试

sysbench 0.5通过一系列LUA脚本来替换之前的oltp，来模拟更接近真实的基准测试环境。这些测试脚本包含：insert.lua、oltp.lua、parallel_prepare.lua、select_random_points.lua、update_index.lua、delete.luaoltp_simple.lua、select.lua、select_random_ranges.lua、update_non_index.lua，脚本使用方式基本类似。

sysbench 0.5默认使用sbtest库，但是需要自己手工先创建好，也可以使用--mysql-db指定，其他非默认项指定选项：

--mysql-host

--mysql-port

--mysql-socket

--mysql-user

--mysql-password

--mysql-db

--mysql-ssl

prepare

生成表并插入数据，可使用parallel_prepare.lua脚本来并行准备数据。

—db-driver 服务器类型　　mysql | drizzle,默认为mysql

—mysql-table-engine 表存数引擎

—myisam-max-rows MyISAM表MAX_ROWS选项(用于大表)

—oltp-table-count 生成表数量[sbtest1、sbtest2...]

—oltp-table-size 生成表的行数

—oltp-secondary ID列生成二级索引而不是主键

—oltp-auto-inc设置ID列是否自增 on | off，默认为on --oltp-read-only=on

--test=sysbench-0.5/sysbench/tests目录下测试脚本

```
sysbench \
--test=/root/sysbench-master/sysbench/tests/db/oltp.lua  \
--mysql-host=localhost  \
--mysql-port=3306  \
--mysql-user=root  \
--mysql-password=uplooking \
--oltp-table-size=100000 \
--num-threads=8 \
--max-time=10 \
--mysql-db=sbtest \
--max-requests=0 \
--oltp-test-mode=complex \
--report-interval=1 \
--mysql-table-engine=innodb  \
[prepare|run|cleanup]准备/测试/清除
```

```
[root@localhost sysbench]# sysbench --test=/root/sysbench-master/sysbench/tests/db/oltp.lua --mysql-host=localhost --mysql-port=3306 --mysql-user=root --mysql-password=uplooking --oltp-table-size=10000 --num-threads=8 --max-time=10 --mysql-db=sbtest --max-requests=0 --oltp-test-mode=complex --report-interval=1 --mysql-table-engine=innodb prepare
sysbench 1.0 (using bundled LuaJIT 2.1.0-beta2)

Creating table 'sbtest1'...
Inserting 10000 records into 'sbtest1'
Creating secondary indexes on 'sbtest1'...

[root@localhost sysbench]# sysbench --test=/root/sysbench-master/sysbench/tests/db/oltp.lua --mysql-host=localhost --mysql-port=3306 --mysql-user=root --mysql-password=uplooking --oltp-table-size=10000 --num-threads=8 --max-time=10 --mysql-db=sbtest --max-requests=0 --oltp-test-mode=complex --report-interval=1 --mysql-table-engine=innodb run
sysbench 1.0 (using bundled LuaJIT 2.1.0-beta2)

Running the test with following options:
Number of threads: 8
Report intermediate results every 1 second(s)
Initializing random number generator from current time


Initializing worker threads...

Threads started!

[   1s] threads: 8, tps: 39.71, reads: 639.17, writes: 162.63, response time: 363.18ms (95%), errors: 0.00, reconnects:  0.00
[   2s] threads: 8, tps: 45.45, reads: 626.37, writes: 185.74, response time: 376.49ms (95%), errors: 0.00, reconnects:  0.00
[   3s] threads: 8, tps: 53.32, reads: 733.18, writes: 214.31, response time: 272.27ms (95%), errors: 0.00, reconnects:  0.00
[   4s] threads: 8, tps: 43.35, reads: 641.97, writes: 176.17, response time: 356.70ms (95%), errors: 0.00, reconnects:  0.00
[   5s] threads: 8, tps: 36.87, reads: 501.05, writes: 160.51, response time: 458.96ms (95%), errors: 0.00, reconnects:  0.00
[   6s] threads: 8, tps: 35.18, reads: 513.58, writes: 141.71, response time: 397.39ms (95%), errors: 0.00, reconnects:  0.00
[   7s] threads: 8, tps: 40.06, reads: 545.80, writes: 147.22, response time: 520.62ms (95%), errors: 0.00, reconnects:  0.00
[   8s] threads: 8, tps: 54.99, reads: 763.82, writes: 216.95, response time: 240.02ms (95%), errors: 0.00, reconnects:  0.00
[   9s] threads: 8, tps: 41.37, reads: 577.26, writes: 176.06, response time: 331.91ms (95%), errors: 0.00, reconnects:  0.00
[  10s] threads: 7, tps: 57.49, reads: 807.87, writes: 217.62, response time: 240.02ms (95%), errors: 0.00, reconnects:  0.00
OLTP test statistics:
    queries performed:
        read:                            6398
        write:                           1828
        other:                           914
        total:                           9140

    transactions:                        457    (44.97 per sec.)
```

```
    read/write requests:                    8226   (809.37 per sec.)
    other operations:                       914    (89.93 per sec.)
    ignored errors:                         0      (0.00 per sec.)
    reconnects:                             0      (0.00 per sec.)

General statistics:
    total time:                             10.1917s
    total number of events:                 457
    total time taken by event execution: 80.3041s

Latency statistics:
         min:                                    22.82ms
         avg:                                   175.72ms
         max:                                   563.77ms
         approx.  95th percentile:              376.49ms

Threads fairness:
    events (avg/stddev):           57.1250/2.93
    execution time (avg/stddev):   10.0380/0.07

# 我的机器是6G内存，2个cpu，单核，超多线程碾压的时候试一试64和128个线程

[root@localhost sysbench]# sysbench --test=/root/sysbench-master/sysbench/tests/db/oltp.lua --
mysql-host=localhost --mysql-port=3306 --mysql-user=root --mysql-password=uplooking --oltp-table-
size=10000 --num-threads=64 --max-time=30 --mysql-db=sbtest --max-requests=0 --oltp-test-
mode=complex --report-interval=1 --mysql-table-engine=innodb run
sysbench 1.0 (using bundled LuaJIT 2.1.0-beta2)

OLTP test statistics:
    queries performed:
        read:                               21210
        write:                              6007
        other:                              3008
        total:                              30225
    transactions:                           1493   (49.07 per sec.)
    read/write requests:                    27217  (894.57 per sec.)
    other operations:                       3008   (98.87 per sec.)
    ignored errors:                         22     (0.72 per sec.)
    reconnects:                             0      (0.00 per sec.)

General statistics:
    total time:                             30.6492s
    total number of events:                 1493
    total time taken by event execution: 1445.1459s

Latency statistics:
         min:                                    35.38ms
         avg:                                   967.95ms
         max:                                  4147.00ms
         approx.  95th percentile:             2238.47ms

Threads fairness:

    events (avg/stddev):           23.3281/12.34
```

```
    execution time (avg/stddev):   22.5804/11.02

[root@localhost sysbench]# sysbench --test=/root/sysbench-master/sysbench/tests/db/oltp.lua --mysql-host=localhost --mysql-port=3306 --mysql-user=root --mysql-password=uplooking --oltp-table-size=10000 --num-threads=128 --max-time=60 --mysql-db=sbtest --max-requests=0 --oltp-test-mode=complex --report-interval=1 --mysql-table-engine=innodb run

OLTP test statistics:
    queries performed:
        read:                            58086
        write:                           16337
        other:                           8198
        total:                           82621
    transactions:                        4049   (67.05 per sec.)
    read/write requests:                 74423  (1232.47 per sec.)
    other operations:                    8198   (135.76 per sec.)
    ignored errors:                      100    (1.66 per sec.)
    reconnects:                          0      (0.00 per sec.)

General statistics:
    total time:                          60.7499s
    total number of events:              4049
    total time taken by event execution: 2685.2741s

Latency statistics:
         min:                                    15.82ms
         avg:                                   663.19ms
         max:                                  8312.00ms
         approx.  95th percentile:             1973.38ms

Threads fairness:
    events (avg/stddev):           31.6328/45.46
    execution time (avg/stddev):   20.9787/22.98
[root@localhost sysbench]# sysbench --test=/root/sysbench-master/sysbench/tests/db/oltp.lua --mysql-host=localhost --mysql-port=3306 --mysql-user=root --mysql-password=uplooking --oltp-table-size=10000 --num-threads=128 --max-time=60 --mysql-db=sbtest --max-requests=0 --oltp-test-mode=complex --report-interval=1 --mysql-table-engine=innodb cleanup
sysbench 1.0 (using bundled LuaJIT 2.1.0-beta2)

Dropping table 'sbtest1'...
```

如果是多表呢并增加表的大小，情况又会如何呢？

```
[root@localhost sysbench]# sysbench --test=/root/sysbench-master/sysbench/tests/db/oltp.lua --mysql-host=localhost --mysql-port=3306 --mysql-user=root --mysql-password=uplooking --oltp-tables-count=10 --oltp-table-size=100000 --num-threads=128 --max-time=60 --mysql-db=sbtest --max-requests=0 --oltp-test-mode=complex --report-interval=1 --mysql-table-engine=innodb prepare
sysbench 1.0 (using bundled LuaJIT 2.1.0-beta2)

Creating table 'sbtest1'...
Inserting 100000 records into 'sbtest1'
Creating secondary indexes on 'sbtest1'...
Creating table 'sbtest2'...
Inserting 100000 records into 'sbtest2'
Creating secondary indexes on 'sbtest2'...
Creating table 'sbtest3'...
Inserting 100000 records into 'sbtest3'
Creating secondary indexes on 'sbtest3'...
Creating table 'sbtest4'...
Inserting 100000 records into 'sbtest4'
Creating secondary indexes on 'sbtest4'...
Creating table 'sbtest5'...
Inserting 100000 records into 'sbtest5'
Creating secondary indexes on 'sbtest5'...
Creating table 'sbtest6'...
Inserting 100000 records into 'sbtest6'
Creating secondary indexes on 'sbtest6'...
Creating table 'sbtest7'...
Inserting 100000 records into 'sbtest7'
Creating secondary indexes on 'sbtest7'...
Creating table 'sbtest8'...
Inserting 100000 records into 'sbtest8'
Creating secondary indexes on 'sbtest8'...
Creating table 'sbtest9'...
Inserting 100000 records into 'sbtest9'
Creating secondary indexes on 'sbtest9'...
Creating table 'sbtest10'...
Inserting 100000 records into 'sbtest10'
Creating secondary indexes on 'sbtest10'...
[root@localhost sysbench]# sysbench --test=/root/sysbench-master/sysbench/tests/db/oltp.lua --mysql-host=localhost --mysql-port=3306 --mysql-user=root --mysql-password=uplooking --oltp-tables-count=10 --oltp-table-size=100000 --num-threads=130 --max-time=20 --mysql-db=sbtest --max-requests=0 --oltp-test-mode=complex --report-interval=1 --mysql-table-engine=innodb run
sysbench 1.0 (using bundled LuaJIT 2.1.0-beta2)

Running the test with following options:
Number of threads: 130
Report intermediate results every 1 second(s)
Initializing random number generator from current time


Initializing worker threads...

Threads started!


[   1s] threads: 130, tps: 0.99, reads: 1259.55, writes: 11.93, response time: 87.56ms (95%),
```

```
errors: 0.00, reconnects:  0.00
[   2s] threads: 130, tps: 14.04, reads: 619.78, writes: 242.70, response time: 1903.57ms (95%),
errors: 0.00, reconnects:  0.00
[   3s] threads: 130, tps: 66.22, reads: 547.33, writes: 261.00, response time: 2880.27ms (95%),
errors: 0.00, reconnects:  0.00
[   4s] threads: 130, tps: 19.21, reads: 366.95, writes: 62.68, response time: 3911.79ms (95%),
errors: 0.00, reconnects:  0.00
[   5s] threads: 130, tps: 45.63, reads: 876.69, writes: 120.39, response time: 4517.90ms (95%),
errors: 0.00, reconnects:  0.00
[   6s] threads: 130, tps: 38.79, reads: 653.12, writes: 183.46, response time: 5312.73ms (95%),
errors: 0.00, reconnects:  0.00
[   7s] threads: 130, tps: 109.66, reads: 833.38, writes: 366.85, response time: 3982.86ms (95%),
errors: 0.00, reconnects:  0.00
[   8s] threads: 130, tps: 9.94, reads: 611.61, writes: 17.90, response time: 4683.57ms (95%),
errors: 0.00, reconnects:  0.00
[   9s] threads: 130, tps: 4.04, reads: 227.04, writes: 74.67, response time: 4358.09ms (95%),
errors: 0.00, reconnects:  0.00
[  10s] threads: 130, tps: 108.98, reads: 813.88, writes: 425.94, response time: 3574.99ms (95%),
errors: 0.00, reconnects:  0.00
[  11s] threads: 130, tps: 21.48, reads: 660.03, writes: 24.41, response time: 3773.42ms (95%),
errors: 0.00, reconnects:  0.00
[  12s] threads: 130, tps: 41.65, reads: 512.63, writes: 221.12, response time: 3982.86ms (95%),
errors: 0.00, reconnects:  0.00
[  13s] threads: 130, tps: 40.33, reads: 623.56, writes: 256.46, response time: 3267.19ms (95%),
errors: 0.00, reconnects:  0.00
[  14s] threads: 130, tps: 39.68, reads: 460.24, writes: 76.38, response time: 3841.98ms (95%),
errors: 0.00, reconnects:  0.00
[  15s] threads: 130, tps: 37.63, reads: 691.16, writes: 215.86, response time: 4517.90ms (95%),
errors: 0.00, reconnects:  0.00
[  16s] threads: 130, tps: 53.97, reads: 704.44, writes: 166.64, response time: 4055.23ms (95%),
errors: 0.00, reconnects:  0.00
[  17s] threads: 130, tps: 22.56, reads: 305.12, writes: 166.53, response time: 4855.31ms (95%),
errors: 0.00, reconnects:  0.00
[  18s] threads: 130, tps: 1.00, reads: 320.21, writes: 36.14, response time: 4943.53ms (95%),
errors: 0.00, reconnects:  0.00
[  19s] threads: 130, tps: 82.52, reads: 854.02, writes: 259.49, response time: 6026.41ms (95%),
errors: 0.00, reconnects:  0.00
[  20s] threads: 130, tps: 11.85, reads: 332.75, writes: 12.84, response time: 4517.90ms (95%),
errors: 0.00, reconnects:  0.00
[  21s] threads: 130, tps: 52.01, reads: 332.47, writes: 383.46, response time: 5124.81ms (95%),
errors: 0.00, reconnects:  0.00
OLTP test statistics:
    queries performed:
        read:                            12656
        write:                           3616
        other:                           1808
        total:                           18080
    transactions:                        904    (42.53 per sec.)
    read/write requests:                 16272  (765.60 per sec.)
    other operations:                    1808   (85.07 per sec.)
    ignored errors:                      0      (0.00 per sec.)
    reconnects:                          0      (0.00 per sec.)
```

```
General statistics:
    total time:                          21.5991s
    total number of events:              904
    total time taken by event execution: 2713.1958s

Latency statistics:
         min:                                    87.55ms
         avg:                                  3001.32ms
         max:                                  6827.14ms
         approx.  95th percentile:             4855.31ms

Threads fairness:
    events (avg/stddev):           6.9538/0.78
    execution time (avg/stddev):   20.8707/0.35
```

CPU测试

> 使用64位整数，测试计算素数直到某个最大值所需要的时间

`sysbench --test=cpu --cpu-max-prime=2000 run`

查看CPU信息方法,查看物理cpu个数

`grep "physical id" /proc/cpuinfo | sort -u | wc -l` 查看核心数量

`grep "core id" /proc/cpuinfo | sort -u | wc -l` 查看线程数量

`grep "processor" /proc/cpuinfo | sort -u | wc -l`

在sysbench的测试中，--num-threads取值为"线程数量"即可

线程(thread)测试

> 测试线程调度器的性能。对于高负载情况下测试线程调度器的行为非常有用

`sysbench --test=threads --num-threads=64 --thread-yields=100 --thread-locks=2 run`

文件IO性能测试

> 生成需要的测试文件，文件总大小5G，16个并发线程。执行完后会在当前目录下生成一堆小文件

`sysbench --test=fileio --num-threads=16 --file-total-size=5G prepare`

执行测试，指定随机读写模式:

- seqwr顺序写入
- seqrewr顺序重写
- seqrd顺序读取
- rndrd随机读取
- rndwr随机写入
- rndrw混合随机读/写

`sysbench --test=fileio --num-threads=16 --init-rng=on --file-total-size=5G --file-test-mode=rndrw run`

除测试文件

```
sysbench --test=fileio --num-threads=16 --file-total-size=5G cleanup
```

内存测试

> 内存测试测试了内存的连续读写性能。

```
sysbench --test=memory --num-threads=16 --memory-block-size=8192 --memory-total-size=1G run
```

互斥锁**(Mutex)**测试

> 测试互斥锁的性能，方式是模拟所有线程在同一时刻并发运行，并都短暂请求互斥锁X。

```
sysbench --test=mutex --num-threads=16 --mutex-num=1024 --mutex-locks=10000 --mutex-loops=5000 run
```

安装**sysbench-0.4**

源码在线下载地址：http://101.96.10.46/downloads.mysql.com/source/sysbench-0.4.12.10.tar.gz

rpm包下线下载地址：http://rpm.pbone.net/index.php3/stat/4/idpl/31446291/dir/centos7/com/sysbench-0.4.12-12.el7.x8664.rpm.html

源码步骤

数据库**rpm**包安装则直接编译

如果数据库是rpm包安装则直接执行以下步骤，否则在configure的时候需要指定mysql数据库的库libs和includes路径

- tar zxf sysbench-0.4.8.tar.gz
- cd sysbench-0.4.8
- ./configure && make && make install
- strip /usr/local/bin/sysbench

数据库为源码编译

数据库源码编译后路径如下：

数据目录　　/usr/local/mysql

includes目录　　/usr/local/mysql/include

libs目录　　/usr/local/mysql/lib

执行以下步骤：

- tar zxf sysbench-0.4.8.tar.gz
- cd sysbench-0.4.8
- ./configure --with-mysql-includes=/usr/local/mysql/include --with-mysql-libs=/usr/local/mysql/lib && make && make install
- strip /usr/local/bin/sysbench

rpm包安装步骤

依赖包：

mariadb-devel-5.5.44-2.el7.x86_64

postgresql-libs.x86*64 0:9.2.13-1.el*71

```
[root@localhost ~]# rpm -q mariadb-devel
mariadb-devel-5.5.44-2.el7.x86_64
[root@localhost ~]#
[root@localhost ~]# rpm -ivh sysbench-0.4.12-12.el7.x86_64.rpm
warning: sysbench-0.4.12-12.el7.x86_64.rpm: Header V4 RSA/SHA1 Signature, key ID 764429e6: NOKEY
error: Failed dependencies:
    libpq.so.5()(64bit) is needed by sysbench-0.4.12-12.el7.x86_64
[root@localhost ~]# yum localinstall -y sysbench-0.4.12-12.el7.x86_64.rpm
Loaded plugins: langpacks, product-id, search-disabled-repos, subscription-manager
This system is not registered to Red Hat Subscription Management. You can use subscription-
manager to register.
Examining sysbench-0.4.12-12.el7.x86_64.rpm: sysbench-0.4.12-12.el7.x86_64
Marking sysbench-0.4.12-12.el7.x86_64.rpm to be installed
Resolving Dependencies
--> Running transaction check
---> Package sysbench.x86_64 0:0.4.12-12.el7 will be installed
--> Processing Dependency: libpq.so.5()(64bit) for package: sysbench-0.4.12-12.el7.x86_64
--> Running transaction check
---> Package postgresql-libs.x86_64 0:9.2.13-1.el7_1 will be installed
--> Finished Dependency Resolution

Dependencies Resolved

================================================================================================
==================================
 Package                       Arch                Version                       Repository
                               Size
================================================================================================
==================================
Installing:
 sysbench                      x86_64              0.4.12-12.el7                 /sysbench-
0.4.12-12.el7.x86_64               172 k
Installing for dependencies:
 postgresql-libs               x86_64              9.2.13-1.el7_1                _mnt
                              230 k

Transaction Summary
================================================================================================
==================================
Install  1 Package (+1 Dependent package)

Total size: 402 k
Total download size: 230 k
Installed size: 836 k
Downloading packages:
Running transaction check
Running transaction test
Transaction test succeeded
Running transaction
  Installing : postgresql-libs-9.2.13-1.el7_1.x86_64
                                1/2


  Installing : sysbench-0.4.12-12.el7.x86_64
```

```
                                2/2
  Verifying  : sysbench-0.4.12-12.el7.x86_64
                                1/2
  Verifying  : postgresql-libs-9.2.13-1.el7_1.x86_64
                                2/2

Installed:
  sysbench.x86_64 0:0.4.12-12.el7


Dependency Installed:
  postgresql-libs.x86_64 0:9.2.13-1.el7_1


Complete!
```

主要测试方式					
cpu性能	磁盘io性能	调度程序性能	内存分配及传输速度	POSIX线程性能	数据库性能(OLTP基准测试)
找范围内最大素数{时间越短越好}	不同场景下IOPS{越大越好}	线程并发执行，循环响应信号量花费的时间{越少越好}	以不同块大小传输一定数量的数据吞吐量大小{越大越好}	并发线程同时申请互斥锁循环一定次数花费的时间{越少越好}	qps、tps越高越好

开始测试

编译成功之后，就要开始测试各种性能了，测试的方法官网网站上也提到一些，但涉及到 OLTP 测试的部分却不够准确。在这里我大致提一下：

cpu性能测试

> 进行素数的加法运算:指定最大的素数为 20000，记录测试结果（可以根据机器cpu的性能来适当调整数值）

sysbench --test=cpu --cpu-max-prime=20000 run

```
[root@localhost ~]# sysbench --test=cpu --cpu-max-prime=20000 run
sysbench 0.4.12:  multi-threaded system evaluation benchmark

Running the test with following options:
Number of threads: 1

Doing CPU performance benchmark

Threads started!
Done.

Maximum prime number checked in CPU test: 20000


Test execution summary:
    total time:                          30.7005s
    total number of events:              10000
    total time taken by event execution: 30.6870
    per-request statistics:
         min:                                  2.47ms
         avg:                                  3.07ms
         max:                                106.81ms
         approx.  95 percentile:               3.74ms

Threads fairness:
    events (avg/stddev):           10000.0000/0.00
    execution time (avg/stddev):   30.6870/0.00
```

线程测试

> 测试64个线程
> sysbench --test=threads --num-threads=64 --thread-yields=100 --thread-locks=2 run

```
sysbench 0.4.12:  multi-threaded system evaluation benchmark

Running the test with following options:
Number of threads: 64

Doing thread subsystem performance test
Thread yields per test: 100 Locks used: 2
Threads started!
Done.


Test execution summary:
    total time:                          1.3949s
    total number of events:              10000
    total time taken by event execution: 66.0646
    per-request statistics:
         min:                                  0.03ms
         avg:                                  6.61ms
         max:                               1265.23ms
         approx.  95 percentile:               9.50ms

Threads fairness:
    events (avg/stddev):           156.2500/302.22
    execution time (avg/stddev):   1.0323/0.33
```

磁盘IO性能测试

> 最大创建16个线程，创建的文件总大小为3G，文件读写模式为随机读

sysbench --test=fileio --num-threads=16 --file-total-size=3G --file-test-mode=rndrw prepare #创建测试文件

sysbench --test=fileio --num-threads=16 --file-total-size=3G --file-test-mode=rndrw run #执行测试文件

sysbench --test=fileio --num-threads=16 --file-total-size=3G --file-test-mode=rndrw cleanup #删除测试文件

```
[root@localhost ~]# sysbench --test=fileio --num-threads=16 --file-total-size=3G --file-test-
mode=rndrw prepare
sysbench 0.4.12:  multi-threaded system evaluation benchmark

128 files, 24576Kb each, 3072Mb total
Creating files for the test...
[root@localhost ~]# sysbench --test=fileio --num-threads=16 --file-total-size=3G --file-test-
mode=rndrw run
sysbench 0.4.12:  multi-threaded system evaluation benchmark

Running the test with following options:
Number of threads: 16

Extra file open flags: 0
128 files, 24Mb each
3Gb total file size
Block size 16Kb
Number of random requests for random IO: 10000
Read/Write ratio for combined random IO test: 1.50
Periodic FSYNC enabled, calling fsync() each 100 requests.
Calling fsync() at the end of test, Enabled.
Using synchronous I/O mode
Doing random r/w test
Threads started!
Done.

Operations performed:  6008 Read, 3997 Write, 12800 Other = 22805 Total
Read 93.875Mb  Written 62.453Mb  Total transferred 156.33Mb  (2.9644Mb/sec)
  189.72 Requests/sec executed

Test execution summary:
    total time:                          52.7348s
    total number of events:              10005
    total time taken by event execution: 434.9929
    per-request statistics:
         min:                                  0.01ms
         avg:                                 43.48ms
         max:                               1105.16ms
         approx.  95 percentile:             180.46ms

Threads fairness:
    events (avg/stddev):           625.3125/46.52
    execution time (avg/stddev):   27.1871/1.19

[root@localhost ~]# sysbench --test=fileio --num-threads=16 --file-total-size=3G --file-test-
mode=rndrw cleanup
sysbench 0.4.12:  multi-threaded system evaluation benchmark

Removing test files...
```

内存测试

> 测试过程是在内存中传输 4G 的数据量，每个 block 大小为 8K。

sysbench --test=memory --memory-block-size=8k --memory-total-size=4G run

OLTP测试

> 测试的表存储引擎类型为innodb，表最大记录数为 1000000，

测试 OLTP 时，先创建数据库 sbtest，或者自己用参数 --mysql-db 来指定其他数据库。

--test=oltp 制定测试类型为OLTP

--db-driver=mysql 测试的数据库类型为mysql

--mysql-table-engine 指定创建的测试表sbtest为 innodb 储引擎类型

--mysql-host=localhost --mysql-user=root --mysql-password=uplooking 分别为服务器ip，用户名和密码

--oltp-table-size=10000 创建的测试表的大小为1万行

--num-threads=128 线程数量为128

```
[root@localhost ~]# sysbench --test=oltp --db-driver=mysql --mysql-host=localhost --mysql-user=root --mysql-password=uplooking --mysql-db=sbtest --mysql-table-engine=innodb --oltp-table-size=10000 --num-threads=128 prepare
sysbench 0.4.12:  multi-threaded system evaluation benchmark

Creating table 'sbtest'...
Creating 10000 records in table 'sbtest'...
[root@localhost ~]# sysbench --test=oltp --db-driver=mysql --mysql-host=localhost --mysql-user=root --mysql-password=uplooking --mysql-db=sbtest --mysql-table-engine=innodb --oltp-table-size=10000 --num-threads=128 run
sysbench 0.4.12:  multi-threaded system evaluation benchmark

Running the test with following options:
Number of threads: 128

Doing OLTP test.
Running mixed OLTP test
Using Special distribution (12 iterations,  1 pct of values are returned in 75 pct cases)
Using "BEGIN" for starting transactions
Using auto_inc on the id column
Maximum number of requests for OLTP test is limited to 10000
Threads started!
Done.

OLTP test statistics:
    queries performed:
        read:                            181986
        write:                           57056
        other:                           22999
        total:                           262041
    transactions:                        10000  (50.00 per sec.)
    deadlocks:                           2999   (14.99 per sec.)
    read/write requests:                 239042 (1195.09 per sec.)
    other operations:                    22999  (114.98 per sec.)

Test execution summary:
    total time:                          200.0198s
    total number of events:              10000
    total time taken by event execution: 25451.6909
    per-request statistics:
         min:                                  9.47ms
         avg:                               2545.17ms
         max:                              26148.89ms
         approx.  95 percentile:            6902.69ms

Threads fairness:
    events (avg/stddev):           78.1250/7.68
    execution time (avg/stddev):   198.8413/0.48

[root@localhost ~]# sysbench --test=oltp --db-driver=mysql --mysql-host=localhost --mysql-user=root --mysql-password=uplooking --mysql-db=sbtest --mysql-table-engine=innodb --oltp-table-size=10000 --num-threads=128 cleanup

sysbench 0.4.12:  multi-threaded system evaluation benchmark
```

```
Dropping table 'sbtest'...
Done.
```

50.00 per sec 为每秒事务量，1195.09 per sec 每秒的读写请求数，total time: 200.0198s 总的用时