#!/usr/bin/env python

# -\*- encoding: utf8 -\*-

# Author: iamacnhero@gmail.cn

# Created: 2009-12-14

from \_\_future\_\_ import division

import MySQLdb, random, datetime, time, os, sys

import ConfigParser

import string, os, sys

# read mysql config file

config\_file = "/root/.my.cnf" #目前应用/etc/my.cnf文件

if (os.path.exists(config\_file)):

cf = ConfigParser.ConfigParser()

cf.read(config\_file)

host = cf.get("client", "host")

user = cf.get("client", "user")

password = cf.get("client", "password")

db = ""

else:

host = '192.168.200.225'

user = 'root'

password = 'p0o9i8u7'

db = ''

# today = time.strftime("%Y-%m-%d %H:%M:%S", time.localtime())

#----------------------------------------------------------------------

def getConn(host, user, passwd, db='jobmd\_new', port=3306, charset=''):

try:

conn = MySQLdb.connect(host=host, user=user, passwd=passwd, db=db, port=port, charset=charset)

return conn

except MySQLdb.Error, e:

print "Error %d: %s" % (e.args[0], e.args[1])

sys.exit(1)

#----------------------------------------------------------------------

def closeConn(conn):

"""close mysql connection"""

conn.close()

#----------------------------------------------------------------------

def getValue(conn, query):

""" get value of query """

cursor = conn.cursor()

cursor.execute(query)

result = cursor.fetchone()

return int(result[1])

def getQuery(conn, query):

""" get more queries """

cursor = conn.cursor()

cursor.execute(query)

result = cursor.fetchall()

return result

Questions = "show global status like 'Questions'"

Uptime = "show global status like 'Uptime'"

Com\_commit = "show global status like 'Com\_commit'"

Com\_rollback = "show global status like 'Com\_rollback'"

Key\_reads = "show global status like 'Key\_reads'"

Key\_read\_requests = "show global status like 'Key\_read\_requests'"

Key\_writes = "show global status like 'Key\_writes'"

Key\_write\_requests = "show global status like 'Key\_write\_requests'"

Have\_innodb = "show global variables like 'have\_innodb'"

Innodb\_buffer\_pool\_reads = "show global status like 'Innodb\_buffer\_pool\_reads'"

Innodb\_buffer\_pool\_read\_requests = "show global status like 'Innodb\_buffer\_pool\_read\_requests'"

Qcache\_hits = "show global status like 'Qcache\_hits'"

Qcache\_inserts = "show global status like 'Qcache\_inserts'"

Open\_tables = "show global status like 'Open\_tables'"

Opened\_tables = "show global status like 'Opened\_tables'"

Threads\_created = "show global status like 'Threads\_created'"

Connections = "show global status like 'Connections'"

Com\_select = "show global status like 'Com\_select'"

Com\_insert = "show global status like 'Com\_insert'"

Com\_update = "show global status like 'Com\_update'"

Com\_delete = "show global status like 'Com\_delete'"

Com\_replace = "show global status like 'Com\_replace'"

Table\_locks\_waited = "show global status like 'Table\_locks\_waited'"

Table\_locks\_immediate = "show global status like 'Table\_locks\_immediate'"

Created\_tmp\_tables = "show global status like 'Created\_tmp\_tables'"

Created\_tmp\_disk\_tables = "show global status like 'Created\_tmp\_disk\_tables'"

Slow\_queries = "show global status like 'Slow\_queries'"

Select\_full\_join = "show global status like 'Select\_full\_join'"

if \_\_name\_\_ == "\_\_main\_\_":

conn = getConn(host, user, password, db)

Questions = getValue(conn, Questions)

Uptime = getValue(conn, Uptime)

Com\_commit = getValue(conn, Com\_commit)

Com\_rollback = getValue(conn, Com\_rollback)

Key\_reads = getValue(conn, Key\_reads)

Key\_read\_requests = getValue(conn, Key\_read\_requests)

Key\_writes = getValue(conn, Key\_writes)

Key\_write\_requests = getValue(conn, Key\_write\_requests)

Qcache\_hits = getValue(conn, Qcache\_hits)

Qcache\_inserts = getValue(conn, Qcache\_inserts)

Open\_tables = getValue(conn, Open\_tables)

Opened\_tables = getValue(conn, Opened\_tables)

Threads\_created = getValue(conn, Threads\_created)

Connections = getValue(conn, Connections)

Com\_select = getValue(conn, Com\_select)

Com\_insert = getValue(conn, Com\_insert)

Com\_update = getValue(conn, Com\_update)

Com\_delete = getValue(conn, Com\_delete)

Com\_replace = getValue(conn, Com\_replace)

Table\_locks\_immediate = getValue(conn, Table\_locks\_immediate)

Table\_locks\_waited = getValue(conn, Table\_locks\_waited)

Created\_tmp\_tables = getValue(conn, Created\_tmp\_tables)

Created\_tmp\_disk\_tables = getValue(conn, Created\_tmp\_disk\_tables)

Slow\_queries = getValue(conn, Slow\_queries)

Select\_full\_join = getValue(conn, Select\_full\_join)

print "\_\_\_\_\_Gerneral Information\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_"

# QPS = Questions / Seconds

QPS = str(round(Questions / Uptime, 5))

print "QPS (Query per seconds): " + QPS

# TPS = (Com\_commit + Com\_rollback ) / Seconds

TPS = str(round((Com\_commit + Com\_rollback)/Uptime, 5))

print "TPS(Transactin per seconds): " + TPS

# Read/Writes Ratio

rwr = str(round((Com\_select + Qcache\_hits) / (Com\_insert + Com\_update + Com\_delete + Com\_replace) \* 100, 5)) + "%"

print "Read/Writes Ratio: " + rwr + "\n"

print "\_\_\_\_\_Cache Usage\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_"

# Key\_buffer\_read\_hits = (1 - Key\_reads / Key\_read\_requests) \* 100%

# Key\_buffer\_write\_hits = (1 - Key\_writes / Key\_write\_requests) \* 100%

Key\_buffer\_read\_hits = str(round((1 - Key\_reads/Key\_read\_requests) \* 100, 5)) + "%"

Key\_buffer\_write\_hits = str(round((1 - Key\_writes/Key\_write\_requests) \* 100, 5)) + "%"

print "MyISAM Key buffer read ratio(99.3% - 99.9% target): " + str(Key\_buffer\_read\_hits)

print "MyISAM Key buffer write ratio: " + str(Key\_buffer\_write\_hits) + "\n"

# Query\_cache\_hits = (Qcache\_hits / (Qcache\_hits + Qcache\_inserts)) \* 100%

Query\_cache\_hits = str(round(((Qcache\_hits/(Qcache\_hits + Qcache\_inserts)) \* 100), 5)) + "%"

print "Query cache hits ratio: " + Query\_cache\_hits + "\n"

cursor = conn.cursor()

cursor.execute(Have\_innodb)

result = cursor.fetchone()

Have\_innodb = result[1]

if (Have\_innodb == "YES"):

Innodb\_buffer\_pool\_reads = getValue(conn, Innodb\_buffer\_pool\_reads)

Innodb\_buffer\_pool\_read\_requests = getValue(conn, Innodb\_buffer\_pool\_read\_requests)

# Innodb\_buffer\_read\_hits = (1 - Innodb\_buffer\_pool\_reads / Innodb\_buffer\_pool\_read\_requests) \* 100%

Innodb\_buffer\_read\_hits = str(round((1 - Innodb\_buffer\_pool\_reads/Innodb\_buffer\_pool\_read\_requests) \* 100, 5)) + "%"

print "Innodb buffer read ratio(target 96% - 99%): " + Innodb\_buffer\_read\_hits + "\n"

# Thread\_cache\_hits = (1 - Threads\_created / Connections) \* 100%

Thread\_cache\_hits = str(round(((1 - Threads\_created / Connections)) \* 100, 5)) + "%"

print "Thread\_cache\_hits(Should above 90%): " + Thread\_cache\_hits + "\n"

print "\_\_\_\_\_Slow Queries(Evil Queries)\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_"

Slow\_queries\_per\_second = str(round(Slow\_queries / (Uptime/60), 5))

print "Slow queries per minute: " + Slow\_queries\_per\_second

Select\_full\_join\_per\_second = str(round(Select\_full\_join / (Uptime/60), 5))

print "Slow full join queries per minute: " + Select\_full\_join\_per\_second

full\_select\_in\_all\_select = str(round((Select\_full\_join / Com\_select) \* 100, 5)) + "%"

print "Full join select queries in all select queries: " + full\_select\_in\_all\_select

# MyISAM Lock Contention: (Table\_locks\_waited / Table\_locks\_immediate) \* 100%

lock\_contention = str(round((Table\_locks\_waited / Table\_locks\_immediate) \* 100, 5)) + "%"

print "MyISAM Lock Contention(<1% good, 1% warning, >3% you are currently dying): " + lock\_contention + "\n"

print "Open tables: " + str(Open\_tables)

print "Opened tables: " + str(Opened\_tables) + "\n"

Temp\_tables\_to\_disk = str(round((Created\_tmp\_disk\_tables / Created\_tmp\_tables) \* 100, 5)) + "%"

print "Temp tables to Disk ratio: " + Temp\_tables\_to\_disk

closeConn(conn)

#说明

QPS (Query per second) （每秒查询量）

TPS(Transaction per second) （每秒事务量，如果是InnoDB会显示，没有InnoDB就不会显示）

Read/Writes Ratio（数据库读写比率，对是否使用MySQL Replication还是使用MySQL Cluster很有参考价值。）

MyISAM Key buffer read ratio

MyISAM Key buffer write ratio

Slow queries per minute （平均一分钟多少慢查询）

Slow full join queries per minute（慢查询的比率）

Temp tables to Disk ratio （写到硬盘的临时表与所有临时表的比率，对性能有较大影响，说明有SQL使用了大量临时表）