# mysql实例运行状态报告脚本

#!/usr/bin/python  
# -\*- coding: utf-8 -\*-  
import commands  
import time  
import re  
import sys  
import MySQLdb

def DB\_SQL(host,user,port,passwd,Time):  
    port1=int(port)  
 password=passwd  
 count=60\*int(Time)  
 print count  
 run\_count=0;  
 cmd="show global status;"  
 conn = MySQLdb.connect (host,user,port=port1,passwd=password)  
 cursor = conn.cursor()  
 f= open("DB\_STATUS\_RECORD.txt",'wr')  
 while run\_count<count:  
  cursor.execute (cmd)  
  rows = cursor.fetchall()  
  for row in rows:  
                 col\_name="%s" %(row[0])  
                    row\_name="%s" %(row[1])   
     #print col\_name, row\_name  
                 if col\_name=='Com\_delete':  
                         all\_sql=row\_name+" "  
                 elif col\_name=='Com\_insert':  
                         all\_sql=all\_sql+row\_name+" "  
                 elif col\_name=='Com\_select':  
                         all\_sql=all\_sql+row\_name+" "  
                 elif col\_name=='Com\_update':  
                         all\_sql=all\_sql+row\_name+" "  
                 elif col\_name=='Slow\_queries':  
                         all\_sql=all\_sql+row\_name+" "  
                 elif col\_name=='Threads\_cached':  
                         all\_sql=all\_sql+row\_name+" "  
                 elif col\_name=='Threads\_connected':  
                         all\_sql=all\_sql+row\_name+" "  
                 elif col\_name=='Threads\_created':  
                         all\_sql=all\_sql+row\_name+" "  
                 elif col\_name=='Threads\_running':  
                         all\_sql=all\_sql+row\_name+" "  
                 elif col\_name=='Uptime':  
                         all\_sql=all\_sql+row\_name+" "  
   else:  
                         continue  
         f.write(all\_sql+"\n")  
  time.sleep(1)  
  run\_count=run\_count+1  
  print ' run\_count=run\_count+1', run\_count  
 cursor.close()  
 conn.close ()  
 f.close()

def statistics\_delete(all\_f):  
        list=[]  
        status,output=commands.getstatusoutput("cat DB\_STATUS\_RECORD.txt |awk '{print $1}'>statistics\_delete\_tmp\_file")  
        f=open("statistics\_delete\_tmp\_file",'r')  
        for data in f.readlines():  
                list.append(re.sub('\n','',data))  
        count=int(max(list))-int(min(list))  
        s="在指定时间内共删除:"+str(count)  
        all\_f.write("-------------\t在指定时间内,删除统计信息如下:\t-------------"+"\n")  
        all\_f.write(s+"\n")

def statistics\_insert(all\_f):  
        list=[]  
        status,output=commands.getstatusoutput("cat DB\_STATUS\_RECORD.txt |awk '{print $2}'>statistics\_delete\_tmp\_file")  
        f=open("statistics\_delete\_tmp\_file",'r')  
        for data in f.readlines():  
                list.append(re.sub('\n','',data))  
        count=int(max(list))-int(min(list))  
        s="在指定时间内共插入:"+str(count)  
        all\_f.write("-------------\t在指定时间内,插入统计信息如下:\t-------------"+"\n")  
        all\_f.write(s+"\n")

def statistics\_select(all\_f):  
        list=[]  
        status,output=commands.getstatusoutput("cat DB\_STATUS\_RECORD.txt |awk '{print $3}'>statistics\_delete\_tmp\_file")  
        f=open("statistics\_delete\_tmp\_file",'r')  
        for data in f.readlines():  
                list.append(re.sub('\n','',data))  
        count=int(max(list))-int(min(list))  
        s="在指定时间内共查询:"+str(count)  
        all\_f.write("-------------\t在指定时间内,查询统计信息如下:\t-------------"+"\n")  
        all\_f.write(s+"\n")  
def statistics\_update(all\_f):  
        list=[]  
        status,output=commands.getstatusoutput("cat DB\_STATUS\_RECORD.txt |awk '{print $4}'>statistics\_delete\_tmp\_file")  
        f=open("statistics\_delete\_tmp\_file",'r')  
        for data in f.readlines():  
                list.append(re.sub('\n','',data))  
        count=int(max(list))-int(min(list))  
        s="在指定时间内共更新:"+str(count)  
        all\_f.write("-------------\t在指定时间内,更新统计信息如下:\t-------------"+"\n")  
        all\_f.write(s+"\n")  
def statistics\_slow\_query(all\_f):  
        list=[]  
        status,output=commands.getstatusoutput("cat DB\_STATUS\_RECORD.txt |awk '{print $5}'>statistics\_delete\_tmp\_file")  
        f=open("statistics\_delete\_tmp\_file",'r')  
        for data in f.readlines():  
                list.append(re.sub('\n','',data))  
        count=int(max(list))-int(min(list))  
        s="在指定时间内共出现慢查询的次数:"+str(count)  
        all\_f.write("-------------\t在指定时间内,慢查询统计信息如下:\t-------------"+"\n")  
        all\_f.write(s+"\n")  
def statistics\_Thread\_Cache(all\_f):  
        list=[]  
        status,output=commands.getstatusoutput("cat DB\_STATUS\_RECORD.txt |awk '{print $6}'>statistics\_delete\_tmp\_file")  
        f=open("statistics\_delete\_tmp\_file",'r')  
        for data in f.readlines():  
                list.append(re.sub('\n','',data))  
        count=int(min(list))  
        s="在指定时间内线程池的个数:"+str(count)  
        all\_f.write("-------------\t在指定时间内,线程池统计信息如下:    -------------"+"\n")  
        if count ==0:  
                all\_f.write("当前线程池不足,根据实际情况是否需要增添线程池innodb\_thread\_pool\_size\n")  
        else:  
                all\_f.write("当前线程池不需要增加\n")  
        all\_f.write(s+"\n")  
def statistics\_Thread\_Connected(all\_f):  
        list=[]  
        status,output=commands.getstatusoutput("cat DB\_STATUS\_RECORD.txt |awk '{print $7}'>statistics\_delete\_tmp\_file")  
        f=open("statistics\_delete\_tmp\_file",'r')  
        for data in f.readlines():  
                list.append(re.sub('\n','',data))  
        min\_count=int(min(list))  
        max\_count=int(max(list))  
        min\_s="在指定时间内连接的最少线程的个数:"+str(min\_count)  
        max\_s="在指定时间内连接的最多线程的个数:"+str(max\_count)  
        all\_f.write("-------------\t在指定时间内,线程连接数统计信息如下:\t-------------"+"\n")  
        all\_f.write(min\_s+"\n")  
        all\_f.write(max\_s+"\n")  
def statistics\_Thread\_Created(all\_f):  
        list=[]  
        status,output=commands.getstatusoutput("cat DB\_STATUS\_RECORD.txt |awk '{print $8}'>statistics\_delete\_tmp\_file")  
        f=open("statistics\_delete\_tmp\_file",'r')  
        for data in f.readlines():  
                list.append(re.sub('\n','',data))  
        min\_count=int(min(list))  
        max\_count=int(max(list))  
        count=max\_count-min\_count  
        all\_f.write("-------------\tMySQL创建的总线程数统计信息如下:\t-------------"+"\n")  
        all\_f.write("创建的线程总数为:"+str(max\_count)+"\n")  
        if count>0:  
                all\_f.write("为提高性能，可以增大innodb\_thread\_pool\_size\n")  
        else:  
                all\_f.write("该参数没有任何影响,无需调整\n")  
def statistics\_Thread\_Running(all\_f):  
        list=[]  
        status,output=commands.getstatusoutput("cat DB\_STATUS\_RECORD.txt |awk '{print $9}'>statistics\_delete\_tmp\_file")  
        f=open("statistics\_delete\_tmp\_file",'r')  
        for data in f.readlines():  
                list.append(re.sub('\n','',data))  
        min\_count=int(min(list))  
        max\_count=int(max(list))  
        count=max\_count-min\_count  
        all\_f.write("-------------\t正在运行的线程数统计信息如下:\t-------------"+"\n")  
        all\_f.write("运行的最大线程总数为:"+str(max\_count)+"\n")  
def statistics\_Uptime(all\_f):  
        list=[]  
        status,output=commands.getstatusoutput("cat DB\_STATUS\_RECORD.txt |awk '{print $10}'>statistics\_delete\_tmp\_file")  
        f=open("statistics\_delete\_tmp\_file",'r')  
        for data in f.readlines():  
                list.append(re.sub('\n','',data))  
        min\_count=int(min(list))  
        max\_count=int(max(list))  
        all\_f.write("-------------\tMySQL 运行时间统计信息:\t-------------"+"\n")  
        all\_f.write("运行时间为:"+str(max\_count)+"秒"+"\n")

if len(sys.argv)==6:  
 host=sys.argv[1]  
 user=sys.argv[2]  
 port=sys.argv[3]  
 passwd=sys.argv[4]  
 Time=int(sys.argv[5])  
 DB\_SQL(host,user,port,passwd,Time)  
 all\_f=open("MYSQL\_STATUS\_REPORT\_FILE",'wr')  
 statistics\_delete(all\_f)  
 statistics\_insert(all\_f)  
 statistics\_select(all\_f)  
 statistics\_update(all\_f)  
 statistics\_slow\_query(all\_f)  
 statistics\_Thread\_Cache(all\_f)  
 statistics\_Thread\_Connected(all\_f)  
 statistics\_Thread\_Created(all\_f)  
 statistics\_Thread\_Running(all\_f)  
 statistics\_Uptime(all\_f)  
 all\_f.close()  
 print "分析的结果请看当前目录下的:MYSQL\_STATUS\_REPORT\_FILE 文件"  
else:  
 print "输入的参数不符合要求:\n"  
 print "例子，脚本名 数据库IP 数据库用户 数据库端口 数据库密码 执行的时间(1代表1分钟)\n"

使用方法：

     脚本名 数据库IP 数据库用户 数据库端口 数据库密码 执行的时间(1代表1分钟)

     例句：mysql\_status.py 192.168.1.37  baojin.qi password  1

程序功能：

     收集制定时间内的mysql的状态参数：

     在指定时间内，执行了多少次增删改查，多少个线程连接mysql系统及系统运行的时间，搜集信息在MYSQL\_STATUS\_REPORT\_FILE里