mysql服务器状态

# 使用系统变量和状态变量

## 查看和设置系统变量值

### 查看系统变量值

Show [GLOBAL|SESSION] variables like '%%';#默认session

show variables where ariable\_name like '%%' and value …;

### 设置系统变量值

#服务器启动的时候设置系统变量（在配置文件中设置）

#运行时设置系统变量

SET GLOBAL var\_name=value;

SET @@GLOBAL.var\_name=value;

SET [SESSION] var\_name=value;

SET @@[SESSION.]var\_name=value;

可以同时设置多个值：

SET GLOBAL v1=val1,… ,SESSION v3=val3,…;

## 查看系统状态值

SHOW [SESSION| GLOBAL] STATUS

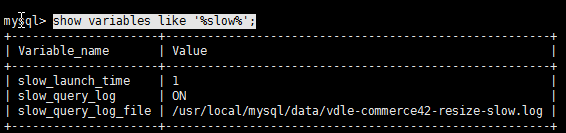
# mysql系统状态和配置调优

系统变量查询show variables like '%%';

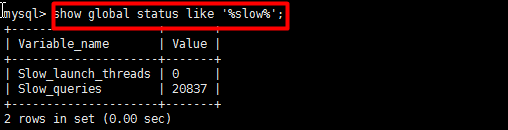
系统状态show global status like '%%';

## 慢查询

show variables like '%slow%';



show global status like '%slow%';

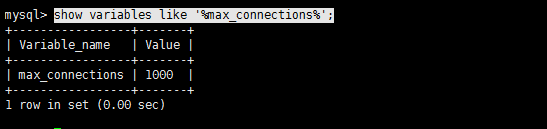


注意：打开慢日志查询功能对系统性能有点影响，可以搭建主从，在其中一台从服务器上打开慢日志监控查询。

## 连接数

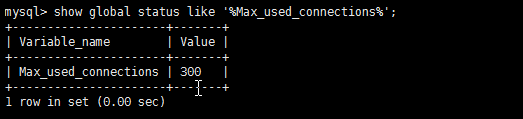
服务器如果报MySQL：ERROR 1040：Too manyconnections的情况。

show variables like '%max\_connections%';



查询服务器过去最大连接数

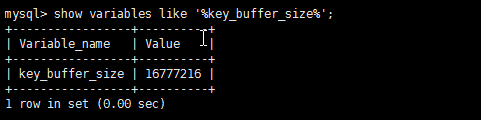
show global status like '%Max\_used\_connections%';



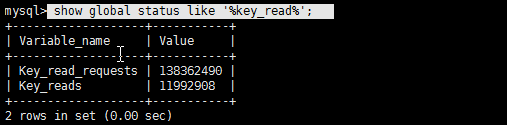
解析：Max\_used\_connections/ max\_connections==85%是比较合理的设置,如果值小于10%，说明mysql连接上线设置的过高。

## key\_buffer\_size

show variables like '%key\_buffer\_size%';



16777216说明分配了16M给key\_buffer\_size，查看key\_buffer\_size的使用情况：show global status like '%key\_read%';

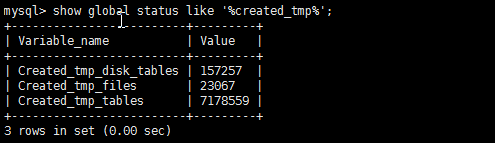


解析：一共有11992908个请求没有在内存在找到，计算索引未命中数：

Key\_cache\_miss\_rate=Key\_reads/ Key\_read\_requests=8.67%，如果Key\_cache\_miss\_rate在0.1%以下，说明很好，如果小于0.01%说明配置过大适当调低。

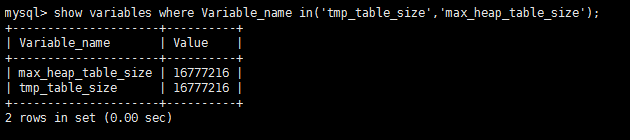
## 临时表

show global status like '%created\_tmp%';



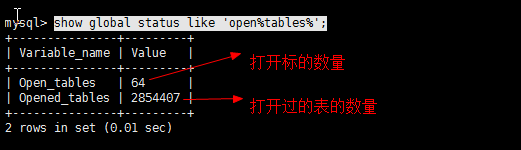
每次创建临时表Created\_tmp\_tables会增加，如果在磁盘创建零时表Created\_tmp\_disk\_tables也会增加。Created\_tmp\_files表示mysql服务器创建的临时文件数。理想配置如下: Created\_tmp\_disk\_tables/Created\_tmp\_disk\_tables<=25%

show variables where Variable\_name in('tmp\_table\_size','max\_heap\_table\_size');

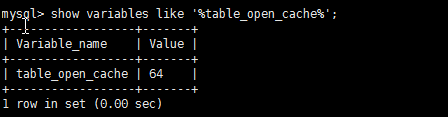


## Open table情况

show global status like 'open%tables%';

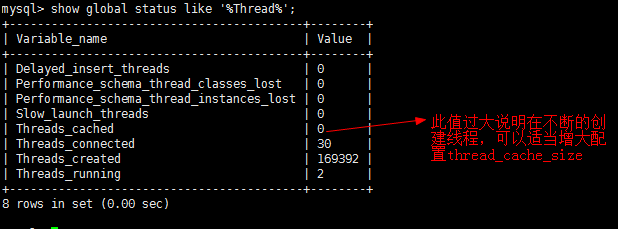


如果Opened\_tables过大说明table\_open\_cache过小

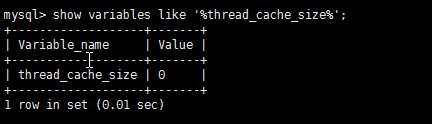


## 线程使用情况

show global status like '%Thread%';

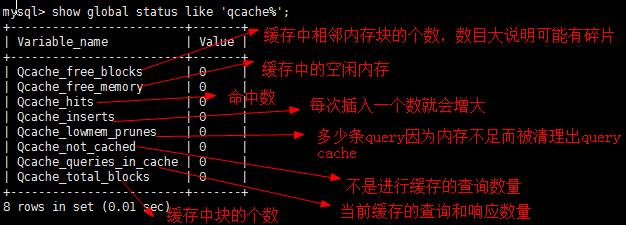


show variables like '%thread\_cache\_size%';

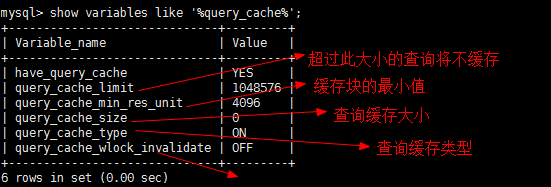


## 查询缓存

show global status like 'qcache%';



Query\_cache的配置：show variables like '%query\_cache%';



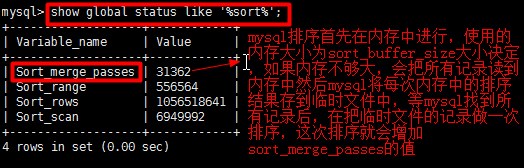
查询缓存碎片率：Qcache\_free\_blocks/Qcache\_total\_blocks\*100%，如果超过了20%，就要执行flush query cache清理

查询缓存利用率：(query\_cache\_size-Qcache\_free\_memory)/ query\_cache\_size\*100%，如果查询缓存利用率在25%以下，说明query\_cache\_size配置过大；如果查询缓存利用率大于80%且Qcache\_lowmem\_prunes>50说明query\_cache\_size配置过小或碎片太大

查询缓存命中率：(Qcache\_hits-Qcache\_inserts)/Qcache\_hits\*100%

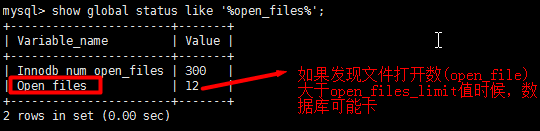
## 排序使用情况

show global status like '%sort%';

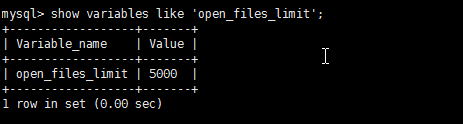


## 文件打开数

show global status like '%open\_files%';



show variables like 'open\_files\_limit';

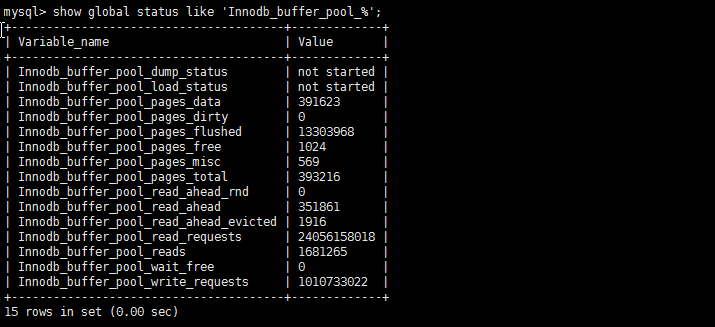


比较合适的设置：Open\_files/open\_files\_limit<=75%

## Innodb\_buffer\_pool\_size的合理设置

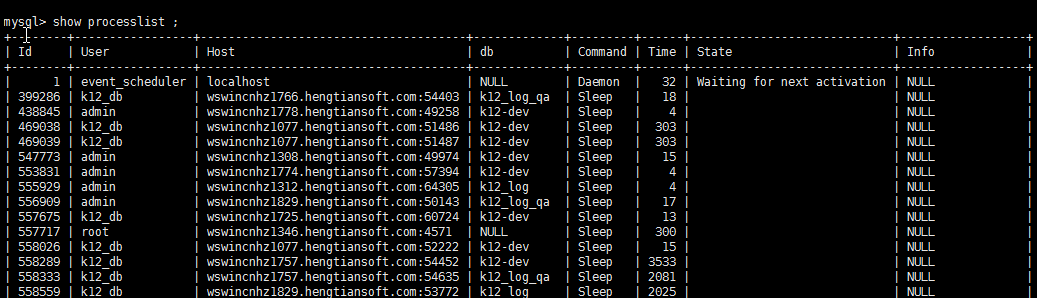
InnoDB引擎的缓存机制，不仅仅缓存索引，同时还会缓存实际的数据。Innodb\_buffer\_pool\_size缓存用户表和索引数据的缓存空间，对InnoDB整体性能影响也最大。

show global status like 'Innodb\_buffer\_pool\_%';



# mysql数据库进程列表

show processlist;



## 说明各列的含义:

* id列:一个标识，你要kill一个语句的时候很有用。
* user列: 显示当前用户，如果不是root，这个命令就只显示你权限范围内的sql语句。
* host列:显示这个语句是从哪个ip的哪个端口上发出的。可用来追踪出问题语句的用户。
* db列:显示这个进程目前连接的是哪个数据库。
* command列:显示当前连接的执行的命令，一般就是休眠（sleep），查询（query），连接（connect）。通常代表资源未释放，如果是通过连接池，sleep状态应该恒定在一定数量范围内
* time列:此这个状态持续的时间，单位是秒。
* state列:显示使用当前连接的sql语句的状态，很重要的列，后续会有所有的状态的描述，请注意，state只是语句执行中的某一个状态，一个sql语句，已查询为例，可能需要经过copying to tmp table，Sorting result，Sending data等状态才可以完成。
* info列:显示这个sql语句，因为长度有限，所以长的sql语句就显示不全，但是一个判断问题语句的重要依据。

## 执行状态分析

**Sleep**状态：通常代表资源未释放，如果是通过连接池，sleep状态应该恒定在一定数量范围内。

**Waiting for net, reading from net, writing to net**状态：偶尔出现无妨，如大量出现，迅速检查数据库到前端的网络连接状态和流量。

**Locked**状态： 有更新操作锁定， 通常使用innodb可以很好的减少locked状态的产生，但是切记，更新操作要正确使用索引，即便是低频次更新操作也不能疏忽。如上影响结果集范例所示。在myisam的时代，locked是很多高并发应用的噩梦。所以mysql官方也开始倾向于推荐innodb。

**Copy to tmp table**： 索引及现有结构无法涵盖查询条件，才会建立一个临时表来满足查询要求，产生巨大的恐怖的i/o压力。很可怕的搜索语句会导致这样的情况，如果是数据分析，或者半夜的周期数据清理任务，偶尔出现，可以允许。频繁出现务必优化之。 Copy to tmp table通常与连表查询有关，建议逐渐习惯不使用连表查询。

**Sending data**： Sending data并不是发送数据，别被名字所欺骗，这是从物理磁盘获取数据的进程，如果你的影响结果集较多，那么就需要从不同的磁盘碎片去抽取数据，偶尔出现该状态连接无碍。一般而言，如果sending data连接过多，通常是某查询的影响结果集过大，也就是查询的索引项不够优化。

**Storing result to query cache**：出现这种状态，如果频繁出现，使用set profiling分析，如果存在资源开销在SQL整体开销的比例过大（即便是非常小的开销，看比例），则说明query cache碎片较多使用flush query cache可即时清理，也可以做成定时任务，Query cache参数可适当酌情设置。

**Freeing items**：理论上这玩意不会出现很多。偶尔出现无碍。如果大量出现，内存，硬盘可能已经出现问题。比如硬盘满或损坏。 i/o压力过大时，也可能出现Free items执行时间较长的情况。

**Sorting for…**：和Sending data类似，结果集过大，排序条件没有索引化，需要在内存里排序，甚至需要创建临时结构排序。

**其他**：还有很多状态，遇到了，去查查资料。基本上我们遇到其他状态的阻塞较少，所以不关心

## mysql kill操作

KILL [CONNECTION | QUERY] thread\_id(终止指定服务器线程)

CONNECTION和没有选项一样：终止给定id的线程。

QUERY：终止给定现场正在执行的所有语句，但不终止该线程本身。

注意：mysqladmin kill 可以终止多个线程的id。

例如：kill 2;