<https://www.cnblogs.com/leeSmall/p/9345993.html>

**一、关键配置**

**1. 配置文件的位置**

 MySQL配置文件

 /etc/my.cnf 或者 /etc/my.cnf.d/server.cnf

几个关键的文件：  
.pid文件，记录了进程id  
.sock文件，是内部通信使用的socket接口，比3306快  
.log文件，日志文件  
.cnf或.conf文件，配置文件  
安装目录：basedir  
数据目录：datadir

**2. 在哪里保存你的数据**

 基本配置，指定数据目录，my.cnf或者server.cnf

[mysqld]  
user = mysql  
port = 3306  
socket = /data/3306/mysql.sock，#这里指定了一个特别的连接  
**basedir = /usr/local/mysql**  
**datadir = /data/3306/data**  
[client]  
port = 3306  
socket = /data/3306/mysql.sock，在客户端也要声明它，命令行要用到

**3. 查询缓存要不要开**

**写入频繁的数据库，不要开查询缓存**

query\_cache\_size  
Query\_cache里的数据又怎么处理呢？首先要把Query\_cache和该表相关的语句全部置为失效，然后在写入更新。那么如果Query\_cache非常大，该表的查询结构又比较多，查询语句失效也慢，一个更新或是Insert就会很慢，这样看到的就是Update或是Insert怎么这么慢了。所以在数据库写入量或是更新量也比较大的系统，该参数不适合分配过大。而且在高并发，写入量大的系统，建议把该功能禁掉。  
query\_cache\_limit   
指定单个查询能够使用的缓冲区大小，缺省为1M  
query\_cache\_min\_res\_unit   
默认是4KB，设置值大对大数据查询有好处，但如果你的查询都是小数据查询，就容易造成内存碎片和浪费

说明：禁掉查询缓存的方法就是直接注释掉查询缓存的配置，如#query\_cache\_size=1M, 这样就可以了

**4. 其他需要开的缓存**

**读缓存，线程缓存，排序缓存**

sort\_buffer\_size = 2M  
connection级参数。太大将导致在连接数增高时，内存不足。

max\_allowed\_packet = 32M  
网络传输中一次消息传输量的最大值。系统默认值 为1MB，最大值是1GB，必须设置1024的倍数。

join\_buffer\_size = 2M  
和sort\_buffer\_size一样，该参数对应的分配内存也是每个连接独享

tmp\_table\_size = 256M  
默认大小是 32M。GROUP BY 多不多的问题

max\_heap\_table\_size = 256M  
key\_buffer\_size = 2048M   
索引的缓冲区大小，对于内存在4GB左右的服务器来说，该参数可设置为256MB或384MB。

read\_buffer\_size = 1M   
read\_rnd\_buffer\_size = 16M   
进行排序查询时，MySql会首先扫描一遍该缓冲，以避免磁盘搜索

bulk\_insert\_buffer\_size = 64M   
批量插入数据缓存大小，可以有效提高插入效率，默认为8M

**Innodb缓存**

innodb\_buffer\_pool\_size = 2048M   
只需要用Innodb的话则可以设置它高达 70-80% 的可用内存。一些应用于 key\_buffer 的规则有 ——如果你的数据量不大，并且不会暴增，那么无需把innodb\_buffer\_pool\_size 设置的太大了。

innodb\_additional\_mem\_pool\_size = 16M  
网络传输中一次消息传输量的最大值。系统默认值为1MB，最大值是1GB，必须设置1024的倍数。

innodb\_log\_files\_in\_group = 3  
循环方式将日志文件写到多个文件。推荐设置为3

innodb\_lock\_wait\_timeout = 120  
InnoDB 有其内置的死锁检测机制，能导致未完成的事务回滚。innodb\_file\_per\_table = 0 独享表空间，关闭

**5. 连接数**

open\_files\_limit = 10240  
允许打开的文件数

back\_log = 600   
短时间内的多少个请求可以被存在堆栈中

max\_connections = 3000  
MySQL默认的最大连接数为100，MySQL服务器允许的最大连接数16384

max\_connect\_errors = 6000   
设置每个主机的连接请求异常中断的最大次数，当超过该次数，MYSQL服务器将禁止host的连接请求

thread\_cache\_size = 300  
重新利用保存在缓存中线程的数量

thread\_concurrency = 8  
thread\_concurrency应设为总CPU核数的2倍

thread\_stack = 192K  
每个线程的堆栈大小，默认值足够大，可满足普通操作。可设置范围为128K至4GB，默认为192KB。

**6. 线程池有关参数**

线程池很少配

thread\_handling  
表示线程池模型。

thread\_pool\_size  
表示线程池的group个数，一般设置为当前CPU核心数目。理想情况下，一个group一个活跃的工作线程，达到充分利用CPU的目的。

thread\_pool\_stall\_limit  
用于timer线程定期检查group是否“停滞”，参数表示检测的间隔。

thread\_pool\_idle\_timeout  
当一个worker空闲一段时间后会自动退出，保证线程池中的工作线程在满足请求的情况下，保持比较低的水平。60秒

thread\_pool\_oversubscribe  
该参数用于控制CPU核心上“超频”的线程数。这个参数设置值不含listen线程计数。

threadpool\_high\_prio\_mode  
表示优先队列的模式。

thread\_pool\_max\_threads  
限制线程池最大的线程数，超过将无法再创建更多的线程，默认为100000。

thread\_pool\_high\_prio\_tickets  
最多语序多少次被放入高优先级队列中，默认为4294967295。只有在thread\_pool\_high\_prio\_mode为transactions的时候才有效果

**说明：**

线程处理的最小单位是statement(语句)  
线程池实现在server端，通过创建一定数量的线程服务DB请求，相对于one-conection-per-thread的一个线程服务一个连接的方式，线程池服务的最小单位是语句，即一个线程可以对应多个活跃的连接。

**7. 慢查询日志**

slow\_query\_log   
是否开启慢查询日志，1表示开启，0表示关闭。

log-slow-queries  
旧版（5.6以下版本）MySQL数据库慢查询日志存储路径。可以不设置该参数，系统则会默认给一个缺省的文件host\_name-slow.log

slow-query-log-file  
新版（5.6及以上版本）MySQL数据库慢查询日志存储路径。可以不设置该参数，系统则会默认给一个缺省的文件host\_name-slow.log

long\_query\_time  
慢查询阈值，当查询时间多于设定的阈值时，记录日志。

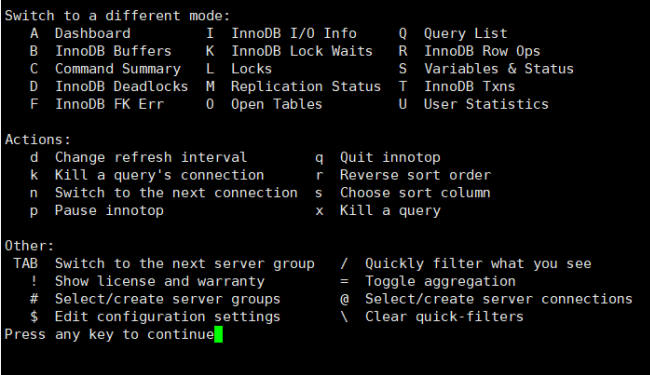
log\_queries\_not\_using\_indexes  
未使用索引的查询也被记录到慢查询日志中（可选项）。

log\_output  
日志存储方式。log\_output='FILE'表示将日志存入文件，默认值是'FILE'。log\_output='TABLE'表示将日志存入数据库，这样日志信息就会被写入到mysql.slow\_log表中。MySQL数据库支持同时两种日志存储方式，配置的时候以逗号隔开即可，如：log\_output='FILE,TABLE'。日志记录到系统的专用日志表中，要比记录到文件耗费更多的系统资源，因此对于需要启用慢查询日志，又需要能够获得更高的系统性能，那么建议优先记录到文件。

**二、监控工具**

**1.  innotop工具**

安装：yum install innotop  
启动：innotop -u root -p ‘123’  
帮助：?



说明：

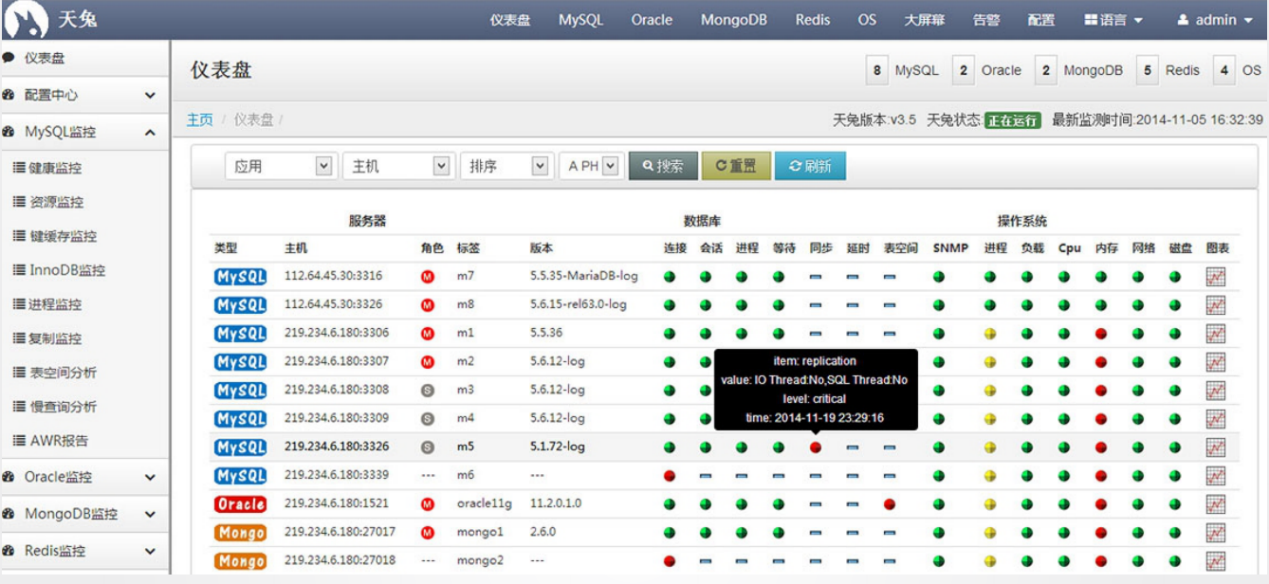
innotop是一个实时工具，只能查看当前的情况，不能记录历史情况

查看帮助的时候输入一个 ？

帮助里面的ABCD.....U就是可以查看对应的指标的情况，如查询列表输入Q、查询缓冲区输入B、命令统计输入C

**2. lepus工具**

http://www.lepus.cc/page/product



说明：

lepus是一款开源的数据库监控工具，能监控各种数据库，能监控多台数据库，能查看历史情况

# <https://cloud.tencent.com/developer/article/1533191>

## [Mysql常用配置及优化](https://links.jianshu.com/go?to=https%3A%2F%2Fwww.cnblogs.com%2Fzhshto%2Fp%2F6653424.html)

## [client]

## 该目录下的内容常用来进行localhost登陆，一般不需要修改

## port = 3306 # 端口号 socket = /var/lib/[mysql](https://cloud.tencent.com/product/cdb?from=10680" \t "_blank)/mysql.sock # 套接字文件(localhost登陆会自动生成)

## [mysql]

## 包含一些客户端mysql命令行的配置

## no-auto-rehash # 默认不自动补全 auto-rehash自动补全

## [mysqld]

## mysql优化的配置目录，除硬件和环境配置外，全部优化在此配置，

## 一般服务器安装只有此配置目录

## user = mysql #默认启动用户，一般不需要修改，可能出现启动不成功 port = 3306 #端口号 socket = /var/lib/mysql/mysql.sock #套接字文件 （套接字方式登陆比TCP/IP方式连接快） character-set-server = utf8 #设置数据库服务器默认编码 utf-8 basedir = /usr/local/mysql #数据库安装目录--指定此参数可解决相对路径造成的问题 datadir = /data/mysql #数据库目录，数据库目录切换时需要用到 pid-file = /var/run/mysqld/mysqld.pid #mysql进程文件，可指定自己的进程文件 external-locking = FALSE #外部锁定(非多服务器可不设置该选项，默认skip-external-locking) skip-external-locking #跳过外部锁定 （避免多进程环境下的external locking--锁机制） skip-name-resolve #跳过主机名解析，直接IP访问，可提升访问速度 log-error = /data/log/mysqld.log #错误日志文件 log-warnings #默认为1,表示启用警告信息记录日志,不需要置0即可,大于1时表示将错误或者失败连接记录日志 open\_files\_limit = 10240 #全局只读变量,文件描述符限制 注：上限其实为OS文件描述符上限，小于OS上限时生效 可用lsof查看限制并修改相应配置

## 以下配置较为重要

## back\_log = 600 #在MYSQL暂时停止响应新请求之前，短时间内的多少个请求可以被存在堆栈中。不超过TCP/IP监听队列，建议512倍数

## 如果系统在短时间内有很多连接，则需要增大该参数的值，该参数值指定到来的TCP/IP连接的监听队列的大小。默认值50。

## max\_connections = 5000 #MySQL允许最大的进程连接数，如果经常出现Too Many Connections的错误提示，则需要增大此值 max\_connect\_errors = 6000 #设置每个主机的连接请求异常中断的最大次数，当超过该次数，MYSQL服务器将禁止host的连接请求，

## 直到mysql服务器重启或通过flush hosts命令清空此host的相关信息

## max\_allowed\_packet = 32M #限制接收数据包的大小,单条数据超过该值时插入或更新失败,

## show VARIABLES like '%max\_allowed\_packet%'; 查看当前限制大小

## sort\_buffer\_size = 8M #每个连接独享，用于优化不能通过sql或者索引优化的group和order等，设置的值应适中

## 比如：500个连接，500\*8 = 4G,将消耗4G内存

## join\_buffer\_size = 8M #用于表间关联缓存的大小,每个连接独享

## thread\_concurrency = 16 #应设为CPU核数的2倍.

## thread\_cache\_size = 600 #线程缓存变量，减少连接的创建，一般设置规则

## 1G->8;2G->16;3G->32;根据实际情况可适当加大(个人：2^(n+2))

## thread\_stack = 1024K #设置MYSQL每个线程的堆栈大小，可设置范围为128K至4GB,默认足够大，一般不用修改 query\_cache\_size = 128M #查询缓存，设置不宜过大，主要是因为缓存越大，设置失效时间越长。如果高并发写，可以禁用该功能 query\_cache\_limit = 2M #指定单个查询能够使用的缓冲区大小，默认为1M query\_cache\_min\_res\_unit = 4k #大数据查询4k,小数据查询2k,目的是提高缓存命中率

## 数据库引擎相关参数

## default-storage-engine = InnoDB #默认数据库引擎 innodb\_data\_file\_path = ibdata1:1024M:autoextend #指定数据文件，初始大小，指定扩展大小 注意与数据库中初始文件大小保持一致

## 格式：size-allocation[:autoextend[:max-size-allocation]]

## innodb\_read\_io\_threads = 16 #read IO线程,根据cpu核心线程数量设置 innodb\_write\_io\_threads = 16 #write IO线程,根据cpu核心线程数量设置 innodb\_thread\_concurrency = 48 #服务器有几个CPU就设置为几，建议用默认设置，一般为8 innodb\_lock\_wait\_timeout = 120 #事务获取锁的最长等待时间，超时请求失败 innodb\_buffer\_pool\_size = 80G #类似于myisam的key\_pool\_size 适当增加可提高命中率，专用服务器可设置为70-80%

## innodb\_flush\_log\_at\_trx\_commit = 2 #默认值1 每次提交日志记录磁盘 2 日志写入系统缓存 0 不提交也记录，不安全，不推荐

## innodb\_flush\_method = O\_DIRECT #控制着innodb数据文件及redo log的打开、刷写模式

## fdatasync(默认)，调用fsync()去刷数据文件与redo log的buffer

## O\_DSYNC，innodb会使用O\_SYNC方式打开和刷写redo log,使用fsync()刷写数据文件

## O\_DIRECT，innodb使用O\_DIRECT打开数据文件，使用fsync()刷写数据文件跟redo log

## fsync() 作用：数作用是flush时将与fd文件描述符所指文件有关的buffer刷写到磁盘

## innodb\_log\_files\_in\_group = 2 #以循环方式将日志文件写到多个文件，默认2 innodb\_log\_file\_size = 4G #数据日志文件大小，较大可提升性能， innodb\_log\_buffer\_size = 512M #日志文件所用的内存大小，以M为单位。缓冲区更大能提高性能，但意外的故障将会丢失数据(开发人员推荐1-8M---不知道为什么)

## innodb\_file\_per\_table = 1 #独立表空间 innodb 默认一个表空间

## innodb\_autoinc\_lock\_mode = 2 #主要作用于自增列

## 0 这个表示tradition 传统 得到语句级别的锁，具有连续性和重复性，但影响并发的插入

## #1 这个表示consecutive 连续 根据一次性插入的数量生成连续的值，

## auto\_inc锁不需要一直保持到语句的结束，语句得到了相应的值后就可以提前释放锁

## #2 这个表示interleaved 交错 没有auto\_inc锁，auto\_incremant值可能不是连续的

## Percona XtraDB Cluster 5.7. 基于同步复制的多主MySQL集群解决方案，只支持Innodb引擎

## 需要安装wsrep打过补丁的版本

## 下载地址：[https://www.percona.com/downloads/](https://links.jianshu.com/go?to=https%3A%2F%2Fwww.percona.com%2Fdownloads%2F)

## pxc\_strict\_mode = ENFORCING wsrep\_cluster\_address = gcomm://192.168.66.242,192.168.66.241 #集群中的节点地址，可以使用主机名或IP wsrep\_node\_address = 192.168.66.241 #本机节点地址，可以使用主机名或IP wsrep\_provider = /usr/lib64/galera3/libgalera\_smm.so wsrep\_sst\_method = xtrabackup-v2 #指定SST方式，支持rsync(最快，需要锁表)，

## mysqldump和xtrabackup，从5.5.33-23.7.6起支持xtrabackup-v2

## wsrep\_slave\_threads = 8 #线程数量 wsrep\_cluster\_name = Cluster #集群名字，必须统一 wsrep\_node\_name = Node1 #集群中节点名字，必须唯一 wsrep\_sst\_auth = "sstuser:8jUthGnAUfWEfJ9" #xtrabackup使用的用户名密码，第一台节点启动时创建，详见安装文档

## wsrep\_sst\_donor='node3,' #一个逗号分割的节点串作为状态转移源，

## 比如wsrep\_sst\_donor=node5,node3, 如果node5可用，用node5,不可用用node3,

## 如果node3不可用，最后的逗号表明让提供商自己选择一个最优的。

## transaction\_isolation = READ-COMMITTED #事务隔离级别

## 1.READ-UNCOMMITTED-读未提交 --脏读(开发生产均不建议)

## 2.READ-COMMITTE-读已提交 --非默认

## 3.REPEATABLE-READ -可重复读 --可能出现幻影行（Innodb 和 Falcon 通过并发解决幻读问题）

## 4.SERIALIZABLE -串行 --最高级别 可能出现大量超时和锁竞争

## tmp\_table\_size = 10G #查询生成的临时表大小超过该值时会在硬盘生成MyISAM表，如果存在大量group by 等语句，可调整大小 max\_heap\_table\_size = 5G #内存表最大空间限制 表在磁盘中，数据在内存中 explicit\_defaults\_for\_timestamp=1 #主要针对TIMESTAMP列，不指定默认会自动加上notnull属性，第一列会自动添加current timestamp table-definition-cache = 4096 #表定义相关信息缓存 实际存放的是frm与内存的映射关系 table-open-cache = 4096 #打开表缓存 存放已打开的表句柄

## 同时设置生效

## interactive\_timeout = 120 #mysql在关闭一个交互的连接之前所要等待的秒数(客户端连接) wait\_timeout = 864000 #mysql在关闭一个非交互的连接之前所要等待的秒数 默认8小时(应用程序调用),

## 根据应用实际情况决定 show processlist; sleep进程多则相应调小

## long\_query\_time = 2 #慢查询超时设置，默认10秒，记录超过查询时间的语句 slow-query-log=1 #注意之前的版本5.6之前的版本为 log-slow-query 是否记录慢查询日志---作为数据库分析 slow-query-log-file=/data/log/query-slow.log #慢查询日志记录文件

## 相关二进制文件设置

## log\_bin #数据库操作二进制记录，数据库备份，复制所需 binlog\_format = ROW #ROW(基于行的复制--安全，但是注意并发) STATEMENT(基于sql语句的复制)，MIXED(混合模式复制) binlog\_cache\_size = 4M #二进制日志缓存，提高log-bin记录效率 log\_bin\_trust\_function\_creators = 1 #主从复制是需要注意，为了保证主从复制完全一致，需要开启此选项，主从默认阻止函数创建 max\_binlog\_size = 1G #二进制日志文件大小默认1G 要求大于4096 小于1G expire\_logs\_days = 7 #清除过期日志

## binlog\_do\_db #此参数表示只记录指定数据库的二进制日志

## binlog\_ignore\_db #忽略某数据库记录

## key\_buffer\_size = 2048M #提高查询命中率，最好不要超过4G,可根据缓存命中率适当调整show global status like 'key\_read%'; read\_buffer\_size = 16M #顺序读缓存，每个连接独享 read\_rnd\_buffer\_size = 64M #随机读缓存，每个连接独享 bulk\_insert\_buffer\_size = 1G #批量插入并且只有向非空表添加数据时才使用该缓存,只对myisam表使用。

## 主从复制相关--必须开启log-bin

## server-id = 19911216 #主从复制必须，并且各服务器具有唯一性 log\_slave\_updates #配置从服务器的更新是否写入二进制日志，默认是不打开的 replicate-ignore-db = mysql #主从复制默认忽略的数据库，可用","分隔或使用多条记录

## replicate-do-db=qrs,login #主从复制指定数据库,","号隔开或使用多条记录

## [mysqldump] #数据库全量备份 quick #强制mysqldump从服务器一次一行地检索表中的行 max\_allowed\_packet = 32M #可接收数据包大小

## [isamchk] #在mysqld服务器不使用的情况下修复表或在崩溃状态下恢复表 key\_buffer = 1024M sort\_buff\_size =1024M read\_buffer = 16M write\_buffer = 16M

## [myisamchk] #在mysqld服务器不使用的情况下修复表或在崩溃状态下恢复表 key\_buffer = 1024M sort\_buff\_size = 1024M read\_buffer = 16M write\_buffer = 16M

## [mysqld\_safe] #safe方式启动数据库，相比于mysqld,会在服务启动后继续监控服务状态，死机时重启 open-files-limit = 8192

# <https://blog.csdn.net/sinat_27933301/article/details/84113002>

[client]

port = 3306

socket = /tmp/mysql.sock

**[mysqld]**

**#Mysql服务的唯一编号 每个mysql服务Id需唯一**

**server-id = 1**

**#服务端口号 默认3306**

**port = 3306**

**#mysql安装根目录**

**basedir = /usr/local/mysql**

**#mysql数据文件所在位置**

**datadir = /usr/local/mysql/data**

**#pid**

**pid-file = /usr/local/mysql/mysql.pid**

**#设置socke文件所在目录**

**socket = /tmp/mysql.sock**

**#设置临时目录**

**tmpdir = /tmp**

**# 用户**

**user = mysql**

**# 允许访问的IP网段**

**bind-address = 0.0.0.0**

**# 跳过密码登录**

**#skip-grant-tables**

**#主要用于MyISAM存储引擎,如果多台服务器连接一个数据库则建议注释下面内容**

**skip-external-locking**

**#只能用IP地址检查客户端的登录，不用主机名**

**skip\_name\_resolve = 1**

**#事务隔离级别，默认为可重复读，mysql默认可重复读级别（此级别下可能参数很多间隙锁，影响性能）**

**transaction\_isolation = READ-COMMITTED**

**#数据库默认字符集,主流字符集支持一些特殊表情符号（特殊表情符占用4个字节）**

**character-set-server = utf8mb4**

**#数据库字符集对应一些排序等规则，注意要和character-set-server对应**

**collation-server = utf8mb4\_general\_ci**

**#设置client连接mysql时的字符集,防止乱码**

**init\_connect='SET NAMES utf8mb4'**

**#是否对sql语句大小写敏感，1表示不敏感**

**lower\_case\_table\_names = 1**

**#最大连接数**

**max\_connections = 400**

**#最大错误连接数**

**max\_connect\_errors = 1000**

**#TIMESTAMP如果没有显示声明NOT NULL，允许NULL值**

**explicit\_defaults\_for\_timestamp = true**

**#SQL数据包发送的大小，如果有BLOB对象建议修改成1G**

**max\_allowed\_packet = 128M**

**#MySQL连接闲置超过一定时间后(单位：秒)将会被强行关闭**

**#MySQL默认的wait\_timeout 值为8个小时, interactive\_timeout参数需要同时配置才能生效**

**interactive\_timeout = 1800**

**wait\_timeout = 1800**

**#内部内存临时表的最大值 ，设置成128M。**

**#比如大数据量的group by ,order by时可能用到临时表，**

**#超过了这个值将写入磁盘，系统IO压力增大**

**tmp\_table\_size = 134217728**

**max\_heap\_table\_size = 134217728**

**#禁用mysql的缓存查询结果集功能**

**#后期根据业务情况测试决定是否开启**

**#大部分情况下关闭下面两项**

**query\_cache\_size = 0**

**query\_cache\_type = 0#数据库错误日志文件**

**log\_error = error.log**

**#慢查询sql日志设置**

**slow\_query\_log = 1**

**slow\_query\_log\_file = slow.log**

**#检查未使用到索引的sql**

**log\_queries\_not\_using\_indexes = 1**

**#针对log\_queries\_not\_using\_indexes开启后，记录慢sql的频次、每分钟记录的条数**

**log\_throttle\_queries\_not\_using\_indexes = 5**

**#作为从库时生效,从库复制中如何有慢sql也将被记录**

**log\_slow\_slave\_statements = 1**

**#慢查询执行的秒数，必须达到此值可被记录**

**long\_query\_time = 8**

**#检索的行数必须达到此值才可被记为慢查询**

**min\_examined\_row\_limit = 100**

**#mysql binlog日志文件保存的过期时间，过期后自动删除**

**expire\_logs\_days = 5**