



链滴

Mysql 参数配置

作者: [zhangxinming](#)

原文链接: <https://ld246.com/article/1537267065861>

来源网站: [链滴](#)

许可协议: [署名-相同方式共享 4.0 国际 \(CC BY-SA 4.0\)](#)

一、配置文件

>配置文件: my.cnf

|My.cnf参数配置||

|----|

|<center>参数|<center>值|<center>描述|

|[client]||客户端读取的参数|

|port|3306|客户端登录使用的端口|

|socket|xx/mysql.scok|客户端使用的socket文件|

|[mysqld]||服务端加载的参数|

|port|3306|服务器设置的端口号|

|socket|xx/mysql.scok|服务器设置的Socket|

|log-bin|mysql-bin|打开二进制功能, 在做主从同步时, 主服务器必须打开
为了安全及基于日
时间点或postion恢复时, 也需要打开|

|log_slave_updates| | 如果从服务作为其他主服务器时 (A->B->C),
需要在B上打开此项.在B上
行的更新会记录到二进制日志.|

|long_query_time|2|执行时长超过此值, 将会认为是慢查询.|

|log_slow_queries| |打开慢查询记录. 超过"long_query_time"的值,
如果 log_long_format
打开,那些没有使用索引的查询也会被记录.|

|log_long_format| | 没有使用索引的查询也被作为到慢速查询附加到慢速日志里|

|back_log|50|操作系统在监听队列中所能保持的连接数,
如出现"connection refused",需要增加
值|

|skip-networking| |不在TCP/IP端口上进行监听,
所有的进程都是在同一台服务器连接到本地的
ysqld,
连接通过Unix sockets 或者命名管道进行,
这样将增强安全系数|

|max_connections|100|服务端同时连接的最大数|

|max_connect_errors|10|每个客户端连接最大的错误允许数量,
如果达到了此限制,将会被MySQL
服务阻止直到执行了"FLUSH HOSTS"
或者服务重启,非法的密码以及其他在链接时的错误会增
此值|

|table_cache |2048|所有线程所打开表的数量,增加此值就增加了mysqld所需要的文件描述符的数量,<
r>这样你需要确认在[mysqld_safe]中 "open-files-limit" 变量设置打开文件数量允许至少4096|

|max_allowed_packet|16M|服务所能处理的请求包的最大大小以及服务所能处理的最大的请求大小|

|binlog_cache_size|1M|在一个事务中binlog为了记录SQL状态所持有的cache大小,
如果你经
使用大的,多声明的事务,
你可以增加此值来获取更大的性能,
所有从事务来的状态都将被缓
在binlog缓冲中然后在提交后一次性写入到binlog中
如果事务比此值大,会使用磁盘上的临时文
来替代,
此缓冲在每个连接的事务第一次更新状态时被创建|

|max_heap_table_size |64M|独立的内存表所允许的最大容量,
此选项为了防止意外创建一个超
的内存表导致永尽所有的内存资源|

|sort_buffer_size |8M|排序缓冲被用来处理类似ORDER BY以及GROUP BY队列所引起的排序

果排序后的数据无法放入排序缓冲, 一个用来替代的基于磁盘的合并分类会被使用
查看 "Sort_me
ge_passes" 状态变量,在排序发生时由每个线程分配|

|join_buffer_size |8M|此缓冲被使用来优化全联合(full JOINS 不带索引的联合),
类似的联合在绝大多数情况下有非常糟糕的性能表现,
但是将此值设大能够减轻性能影响.
通过 "Select_full_join" 状态变量查看全联合的数量
当全联合发生时,在每个线程中分配|

|thread_cache_size |8|我们在cache中保留多少线程用于重用
当一个客户端断开连接后,如果cache中的线程还少于thread_cache_size,
则客户端线程被放入cache中.|

|query_cache_size |64M| 查询缓冲常被用来缓冲 SELECT 的结果并且在下一次同样查询的时候不再行直接返回结果
查看 "Qcache_lowmem_prunes" 状态变量来检查是否当前值对于你的负载来是否足够高
表经常变化的情况下或者如果你的查询原文每次都不同,查询缓冲也许引起性能下降不是性能提升|

|query_cache_limit|2M|只有小于此设定值的结果才会被缓冲,此设置用来保护查询缓冲,防止一个极大结果集将其他所有的查询结果都覆盖.|

|tmp_table_size |64M|内部(内存中)临时表的最大大小,如果一个表增长到比此值更大,将会自动转换基于磁盘的表. 此限制是针对单个表的,而不是总和.|

|InnoDB||存储引擎相关|

|innodb_buffer_pool_size|2G |InnoDB使用一个缓冲池来保存索引和原始数据, 建议服务器物理内存大小的80%|

|innodb_flush_log_at_trx_commit|2|0: 每隔一秒, 把事务日志缓存区的数据写到日志中, 以及把日志文件的数据刷新到磁盘上.
1: 每个事务提交时候, 把事务日志从缓存区写到日志文件中, 并且刷日志文件的数据到磁盘上
2: 每次提交后日志写入日志文件,但只有大约每秒将日志文件刷新到磁盘上.|

|innodb_log_buffer_size|8M|来缓冲日志数据的缓冲区的大小, 当此值快满时, InnoDB将必须刷新数据到磁盘上.
由于基本上每秒都会刷新一次,所以没有必要将此值设置的太大|

|innodb_log_file_size |256M|在日志组中每个日志文件的大小, 文件太小, 频繁刷新, 文件太大, 恢时间长|

二、加载顺序

1.Option Files Read on Windows Systems

File Name	Purpose
%WINDIR%\my.ini, %WINDIR%\my.cnf options	Global options
C:\my.ini, C:\my.cnf	Global options
BASEDIR\my.ini, _BASEDIR_\my.cnf options	Global options
defaults-extra-file ults-extra-file, if any	The file specified with --defaults-extra-file
``%APPDATA%\MySQL\mylogin.cnf options (clients only)	Login path

2.Option Files Read on Unix and Unix-Like Systems

File Name

`/etc/my.cnf`

`/etc/mysql/my.cnf`

`_SYSCONFDIR_/my.cnf`

`$MYSQL_HOME/my.cnf`
(server only)

`defaults-extra-file`
`ults-extra-file`, if any

`~/my.cnf`

`~/mylogin.cnf`
(clients only)

``

Purpose

Global options

Global options

Global options

Server-specific option

The file specified with `--def`

User-specific options

User-specific login path option