

服务器参数

▼ general —— 没啥可以优化的

- `datadir=/var/lib/mysql`
数据文件存放的目录
- `socket=/var/lib/mysql/mysql.sock`——一般不作 修改
`mysql.socket`表示server和client在同一台服务器，并且使用localhost进行连接，就会使用socket进行连接
- `pid_file=/var/lib/mysql/mysql.pid`
存储mysql的pid 进程id
- `port=3306`
mysql服务的端口号
- `default_storage_engine=InnoDB`
mysql存储引擎—— 默认是InnoDB
- `skip-grant-tables`
当忘记mysql的用户名密码的时候，可以在mysql配置文件中配置该参数，跳过权限表验证，不需要密码即可登录mysql

▼ character —— 不同字符占用的空间不同（一般很少去做这件事）

- `character_set_client`
客户端数据的字符集
- `character_set_connection`
mysql处理客户端发来的信息时，会把这些数据转换成连接的字符集格式
- `character_set_results`
mysql发送给客户端的结果集所用的字符集
- `character_set_database`
数据库默认的字符集

- character_set_server

mysql server的默认字符集

▼ connection —— 不建议设置的太大 (因为一般有连接池)

- max_connections

mysql的最大连接数，如果数据库的并发连接请求比较大，应该调高该值，
可以更改，但是和服务器硬件配置相关（默认数量和硬件有关系）

```
mysql> show variables like '%max_connection%'
-> ;
+-----+-----+
| Variable_name | Value |
+-----+-----+
| max_connections | 151 |
+-----+-----+
1 row in set (0.01 sec)
```

修改的时，不要设置的太大

```
mysql> set max_connections=1024;
ERROR 1193 (HY000): Unknown system variable 'max_connections'
mysql> set max_connections=1024;
ERROR 1229 (HY000): Variable 'max_connections' is a GLOBAL variable and should be set with SET GLOBAL
mysql> set global max_connections=1024;
Query OK, 0 rows affected (0.00 sec)

mysql> show variables like '%max_connection%';
+-----+-----+
| Variable_name | Value |
+-----+-----+
| max_connections | 1024 |
+-----+-----+
1 row in set (0.00 sec)
```

- max_user_connections

限制每个用户的连接个数（默认是0，0表示不限制），

可以通过 show variables like '%connection%' 查看

- back_log —— 不宜设置太大，一般不设置

mysql能够暂存的连接数量，当mysql的线程在一个很短时间内得到非常多的连接请求时，就会起作用，如果mysql的连接数量达到max_connections时，新的请求会被存储在堆栈中，以等待某一个连接释放资源，如果等待连接的数量超过back_log,则不再接受连接资源

如最大连接数是100， back_log 暂存数量是80，则 连接数达到100后，会暂存80个，类似于blockingQueue

- wait_timeout
mysql在关闭一个非交互的连接之前需要等待的时长
- interactive_timeout
关闭一个交互连接之前需要等待的秒数



命令行：交互式的（长链接，一直交互）
jdbc是非交互式的（每次打开连接、执行sql、关闭连接）

▼ log

- log_error
指定错误日志文件名称，用于记录当mysqld启动和停止时，以及服务器在运行中发生任何严重错误时的相关信息
- log_bin 默认不开启 —— 一般情况下要开启 log-bin=master-bin
最好开启，因为bin log 中会存有一定的数据日志，如果数据丢失，可以从这个里面恢复
指定二进制日志文件名称，用于记录对数据造成更改的所有查询语句
- binlog_do_db
指定将更新记录到二进制日志的数据库，其他所有没有显式指定的数据库更新将忽略，不记录在日志中，通过指定数据库名称，将指定数据库的日志写入 bin log 日志，这样如果一个mysql中有好多数据库的时候，可以过滤掉一些不重要的db，减少log数据量
- binlog_ignore_db
指定不将更新记录到二进制日志的数据库，同上 相反 白名单和黑名单
- sync_binlog
指定多少次写日志后同步磁盘
写日志的时候有一个log buffer，操作系统有对应的os buffer
- general_log——如果性能允许的情况，建议开启
是否开启查询日志记录，对sql记录进行一个记录备份

```
mysql> show variables like '%general_log%'
-> ;
+-----+-----+
| Variable_name | Value |
+-----+-----+
| general_log   | OFF   |
| general_log_file | /var/lib/mysql/node01.log |
+-----+-----+
2 rows in set (0.01 sec)
```

- `general_log_file`
指定查询日志文件名，用于记录所有的查询语句
- `slow_query_log` —— 一般是开启
是否开启慢查询日志记录
如果一个语句执行的比较慢的话，会记录日志到对应文件中
- `slow_query_log_file`
指定慢查询日志文件名称，用于记录耗时比较长的查询语句
- `long_query_time` —— 满查询 快慢的标准
设置慢查询的时间，超过这个时间的查询语句才会记录日志
- `log_slow_admin_statements`
是否将管理语句写入慢查询日志

▼ [cache](#)

- ▼ `key_buffer_size` 索引缓存区的大小（只对myisam表起作用）
- ▼ query cache（8的时候被淘汰了）
 - `query_cache_size` 查询缓存的大小，未来版本被删除
 1. `show status like '%Qcache%'`;查看缓存的相关属性
 2. `Qcache_free_blocks`：缓存中相邻内存块的个数，如果值比较大，那么查询缓存中碎片比较多
 3. `Qcache_free_memory`：查询缓存中剩余的内存大小
 4. `Qcache_hits`：表示有多少此命中缓存
 5. `Qcache_inserts`：表示多少次未命中而插入
 6. `Qcache_lowmem_prunes`：多少条query因为内存不足而被移除cache

7. Qcache_queries_in_cache：当前cache中缓存的query数量

8. Qcache_total_blocks：当前cache中block的数量

- query_cache_limit 超出此大小的查询将不被缓存
- query_cache_min_res_unit 缓存块最小大小
- query_cache_type 缓存类型，决定缓存什么样的查询
 1. 0 表示禁用
 2. 1 表示将缓存所有结果，除非sql语句中使用sql_no_cache禁用查询缓存
 3. 2 表示只缓存select语句中通过sql_cache指定需要缓存的查询

▼ sort_buffer_size 每个需要排序的线程分派该大小的缓冲区

▼ max_allowed_packet=32M 限制server接受的数据包大小——默认值就够了，很少改

▼ join_buffer_size=2M 表示关联缓存的大小

▼ thread_cache_size

服务器线程缓存，这个值表示可以重新利用保存再缓存中的线程数量，当断开连接时，那么客户端的线程将被放到缓存中以响应下一个客户而不是销毁，如果线程重新被请求，那么请求将从缓存中读取，如果缓存中是空的或者是新的请求，这个线程将被重新请求，那么这个线程将被重新创建，如果有很多新的线程，增加这个值即可

- Threads_cached：代表当前此时此刻线程缓存中有多少空闲线程
- Threads_connected：代表当前已建立连接的数量
- Threads_created：代表最近一次服务启动，已创建现成的数量，如果该值比较大，那么服务器会一直再创建线程
- Threads_running：代表当前激活的线程数

▼ INNODB

- innodb_buffer_pool_size

该参数指定大小的内存来缓冲数据和索引，最大可以设置为物理内存的80%

- innodb_flush_log_at_trx_commit

主要控制innodb将log buffer中的数据写入日志文件并flush磁盘的时间点，值分别为0，1，2

- innodb_thread_concurrency

设置innodb线程的并发数，默认为0表示不受限制，如果要设置建议跟服务器的cpu核心数一致或者是cpu核心数的两倍

- innodb_log_buffer_size

此参数确定日志文件所用的内存大小，以M为单位

- innodb_log_file_size

此参数确定数据日志文件的大小，以M为单位

- innodb_log_files_in_group

以循环方式将日志文件写到多个文件中

- read_buffer_size

mysql读入缓冲区大小，对表进行顺序扫描的请求将分配到一个读入缓冲区

- read_rnd_buffer_size

mysql随机读的缓冲区大小

- innodb_file_per_table

此参数确定为每张表分配一个新的文件