1. use=mysql

此配置添加还需 systemctl stop mysqld 然后再mysqld --user=mysql --explicit\_defaults\_for\_timestamp 不进行此操作会报错

1. table\_open\_cache = 4000 默认 【单位个数】table\_cache(5.1.3之后这个值叫做table\_open\_cache)

使用《参数优化方案》里头方法mysql> show status;查看参数opened\_tables，再通过《mysql配置参数详解》了解参数；然后再mysqld --verbose --help 查看参数table\_open\_cache 对比 ；Table cache hit rate = table\_open\_cache\*100/Opened\_tables；Table cache hit rate 的值保持百分之八十左右是最佳；【当然还有一个对于table\_open\_cache 的参数联动的设置 open\_file\_limit 所以在调整你的table\_open\_cache 时也是需要调整你的open\_file\_limit 可能要调大，open\_files\_limit= Table\_open\_cache\*2】mysqld --validate-config初步验证修改配置时出现语法错误和是否存在此配置等问题；再重启mysqld服务；再通过mysql> SHOW VARIABLES;查看是否修改成功；

1. key\_buffer\_size = 8388608默认 MyISAM表性能影响最大的一个参数，如果没有设置myisam 那么将不生效；

使用《参数优化方案》里头方法mysql> show status;查看参数Key\_reads，再通过《mysql配置参数详解》了解参数；然后再mysqld --verbose --help 查看参数key\_buffer\_size 对比 ；再通过《参数优化方案》里头的方法修改参数还是使用默认值；修改完配置；mysqld --validate-config初步验证修改配置时出现语法错误和是否存在此配置等问题；再重启mysqld服务；再通过mysql> SHOW VARIABLES;查看是否修改成功；

1. thread\_cache\_size = 9 默认

表示可以重新利用保存在缓存中线程的数量; #缓存的最大线程数。当客户端断开之后，服务器处理此客户的线程将会缓存起来以响应下一个客户而不是销毁（前提是缓存数未达上限）。

# 根据物理内存设置规则如下：  
# 1G  —> 8  
# 2G  —> 16  
# 3G  —> 32  
# 大于3G  —> 64

thread\_cache\_size = 64

1. tmp\_table\_size = 16777216 默认 【设置64M时 在SHOW VARIABLES;里查看是67108864 默认是16M 它会自动转换成byte字节单位】

使用《参数优化方案》里头方法mysql> show status;查看参数Created\_tmp\_disk\_table，再通过《mysql配置参数详解》了解参数；然后再mysqld --verbose --help 查看参数tmp\_table\_size 对比 ；再通过《参数优化方案》里头的方法修改参数还是使用默认值；修改完配置；mysqld --validate-config初步验证修改配置时出现语法错误和是否存在此配置等问题；再重启mysqld服务；再通过mysql> SHOW VARIABLES;查看是否修改成功；

1. log-error=/var/log/mysqld.log 默认 【日志】

log-error=/var/lib/mysql/mysqld.log；修改一下方便查找；

修改配置文件中 日志文件地址：移到datadir数据目录中 [使用绝对地址]；然后进行

Shell>touch mysqld.log

Shell>chown mysql:mysql mysqld.log

Shell>chmod 750 mysqld.log

操作 再重启mysqld

1. slow\_query\_log =OFF 默认关闭 【开启慢日志】

使用mysql>show variables like '%slow%';查看参数；slow\_query\_log OFF；slow\_query\_log =1  ;# 1表示开启，0表示关闭。还需要进行和上头一样的操作；

1. slow\_query\_log\_file=/var/lib/mysql/mysql\_slow.log 默认

慢日志的文件地址；

1. long\_query\_time=2  默认

#慢查询时间  超过2秒则为慢查询

1. max\_connections=151  默认 mysql服务器支持的最大并发连接数据;根据以下四点设置；

a. show variables like '% conne%'; ##查询数据库当前设置的最大连接数 'Max\_used\_connections'; ##服务器响应的最大连接数 一般后者是前者的85%

b. 一般情况下根据同时在线人数设置一个比较综合的数字，我们设置的是1000.

c. max\_connections 还取决于操作系统对单进程允许打开最大文件数的限制; MySQL 的 open\_files\_limit 参数值是在MySQL启动时记录的操作系统对单进程打开最大文件数限制的值可以使用 show variables like 'open\_files\_limit'; 默认10000；查看 open\_files\_limit 值

d. 直接在 Linux 下通过ulimit -n命令查看操作系统对单进程打开最大文件数限制 ( 默认为1024 )

每个连接都需要内存。这个数字受你的内存限制。与操作系统无关。内存足够的话，你可以设置1-100000之间的数字。

1. tmp\_table\_size = 16777216 默认 16M 单位字节byte

# 内存中的每个临时表允许的最大大小。如果临时表大小超过该值，临时表将自动转为基于磁盘的表 tmp\_table\_size和max\_heap\_table\_size在数据表的数据量非常大的时候可以适当增加，增加到tmp\_table\_size = 64M，可达到提高联接查询速度的效果但是不可以一直增加，防止多余内存泄漏；mysql > show variables like 'tmp\_table\_size';

1. max\_heap\_table\_size = 16777216 默认 16M 单位字节byte

# MySQL的heap（堆积）表缓冲大小。所有联合在一个DML指令内完成，并且大多数联合甚至可以不用临时表即可以完成。 tmp\_table\_size和max\_heap\_table\_size在数据表的数据量非常大的时候可以适当增加，max\_heap\_table\_size = 64M，可达到提高联接查询速度的效果但是不可以一直增加，防止多余内存泄漏；mysql> show variables like 'max\_heap\_table\_size'；

1. thread\_concurrency = 8 默认

使# 此允许应用程序给予线程系统一个提示在同一时间给予渴望被运行的线程的数量.此值只对于支持 thread\_concurrency() 函数的系统有意义( 例如Sun Solaris).

# 你可可以尝试使用 [CPU数量]\*(2..4) 来作为 thread\_concurrency 的值；该参数取值为服务器逻辑CPU数量×2

1. open\_ files\_limit = 10000 默认

比较max\_connections\*5和ulimit -n的值，哪个大用哪个；一般不需要修改；mysql>show  status like 'open\_files'; Open\_files / open\_files\_limit \* 100% <= 75％

1. connect\_timeout = 10 默认

使用默认值；

1. wait\_timeout = 28800 默认 8小时

wait\_timeout = 9000；2.5个小时；超时等待时间，单位秒，即一个connection在若干秒内无响应，则服务器切断与这个客户端的连接；

1. back\_log = 500 默认

使用默认值；

1. read\_buffer\_size = 131072 默认 0.152M 【可增加到2M】

# MySQL读入缓冲区大小。对表进行顺序扫描的请求将分配一个读入缓冲区，MySQL会为它分配一段内存缓冲区。read\_buffer\_size变量控制这一缓冲区的大小；

1. read\_rnd\_buffer\_size = 262144 默认 0.304M 【可增加到2-4M】

# 随机读取数据缓冲区使用内存(read\_rnd\_buffer\_size)：和顺序读取相对应；

1. sort\_buffer\_size = 262144 默认 0.304M 【可增加到2-4M】

# 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；

1. join\_buffer\_size = 262144 默认 0.304M 【可增加到2-4M】

# 此缓冲被使用来优化全联合(FULL JOINS 不带索引的联合)；