

在Linux高负载下mysql数据库彻底优化二 PDF转换可能丢失图片或格式，建议阅读原文

https://www.100test.com/kao_ti2020/144/2021_2022__E5_9C_A8L

[inux_E9_AB_c103_144167.htm](https://www.100test.com/kao_ti2020/144/2021_2022__E5_9C_A8L_inux_E9_AB_c103_144167.htm) 对于有1G内存的机器，推荐值是128 - 256。笔者设置table_cache = 256 得到以下状态: Open tables 256 Opened tables 9046 虽然open_tables已经等于table_cache，但是相对于服务器运行时间来说,已经运行了20天，opened_tables的值也非常低。因此，增加table_cache的值应该用处不大。如果运行了6个小时就出现上述值 那就要考虑增大table_cache。如果你不需要记录2进制log 就把这个功能关掉，注意关掉以后就不能恢复出问题前的数据了，需要您手动备份，二进制日志包含所有更新数据的语句，其目的是在恢复数据库时用它来把数据尽可能恢复到最后的状态。另外，如果做同步复制(Replication)的话，也需要使用二进制日志传送修改情况。log_bin指定日志文件，如果不提供文件名，MySQL将自己产生缺省文件名。MySQL会在文件名后面自动添加数字引，每次启动服务时，都会重新生成一个新的二进制文件。此外，使用log-bin-index可以指定索引文件；使用binlog-do-db可以指定记录的数据库；使用binlog-ignore-db可以指定不记录的数据库。注意的是：binlog-do-db和binlog-ignore-db一次只指定一个数据库，指定多个数据库需要多个语句。而且，MySQL会将所有的数据库名称改成小写，在指定数据库时必须全部使用小写名字，否则不会起作用。关掉这个功能只需要在他前面加上#号 #log-bin 开启慢查询日志(slow query log) 慢查询日志对于跟踪有问题的查询非常有用。它记录所有查过long_query_time的

查询，如果需要，还可以记录不使用索引的记录。下面是一个慢查询日志的例子：开启慢查询日志，需要设置参数 `log_slow_queries`、`long_query_times`、`log-queries-not-using-indexes`。`log_slow_queries`指定日志文件，如果不提供文件名，MySQL将自己产生缺省文件名。`long_query_times`指定慢查询的阈值，缺省是10秒。`log-queries-not-using-indexes`是4.1.0以后引入的参数，它指示记录不使用索引的查询。笔者设置 `long_query_time=10` 笔者设置: 参数说明: `back_log` 要求MySQL能有的连接数量。当主要MySQL线程在一个很短时间内得到非常多的连接请求，这就起作用，然后主线程花些时间(尽管很短)检查连接并且启动一个新线程。`back_log`值指出在MySQL暂时停止回答新请求之前的短时间内多少个请求可以被存在堆栈中。只有如果期望在一个短时间内有很多连接，你需要增加它，换句话说，这值对到来的TCP/IP连接的侦听队列的大小。你的操作系统在这个队列大小上有它自己的限制。Unix `listen(2)`系统调用的手册页应该有更多的细节。检查你的OS文档找出这个变量的最大值。试图设定`back_log`高于你的操作系统的限制将是无效的。`max_connections` 并发连接数目最大，120 超过这个值就会自动恢复，出了问题能自动解决 `thread_cache` 没找到具体说明，不过设置为32后 20天才创建了400多个线程 而以前一天就创建了上千个线程 所以还是有用的 `thread_concurrency` # 设置为你的cpu数目x2,例如，只有一个cpu,那么 `thread_concurrency=2` #有2个cpu,那么 `thread_concurrency=4`
`skip-innodb` #去掉innodb支持 100Test 下载频道开通，各类考试题目直接下载。详细请访问 www.100test.com