# 样例代码

大家来找茬，欢迎吐槽；头脑风波、发散思维，看看有哪些问题？

复制工程 循环内部执行sql

|  |
| --- |
| for (int i = 0; i < listnewTProTreeEntitys.size(); i++) {  …  //复制表格  List<TTablesEntity> tablesEntity\_old = tableRepository.findAllByTreeId(oldtreeidlist.get(i));  if (null != tablesEntity\_old && 1 <= tablesEntity\_old.size()) {  for(int y = 0; y < tablesEntity\_old.size(); y++){  listnewTTablesEntitys.add(tablesEntity);  table\_ids+="'"+tablesEntity\_old.get(y).getTableId()+"',"; |

# 优化

犯错属于 欠缺最基本的常识，循环内进行io操作。

2000个模板树，会执行2000次取表格的sql

优化后：

|  |
| --- |
| // TODO 循环外一次查询所有数据  List<TTablesEntity> tablesEntity\_old = tableRepository.findAllByProId(projid);  for (int i = 0; i < listnewTProTreeEntitys.size(); i++) {  if (null != tablesEntity\_old && 1 <= tablesEntity\_old.size()) {  for(int y = 0; y < tablesEntity\_old.size(); y++){  // treeid 不等于 原工程的treeid，continue;否则设置 新的treeid，TTablesEntity  if(!oldtreeidlist.get(i).equals(tablesEntity\_old.get(y).getProTreeId())){  continue;  } |

# 对比分析

一分钟到一秒钟的优化

projectTree 表总数据量： 500万 5154575条

|  |  |  |  |
| --- | --- | --- | --- |
|  | 耗时 | 优化后 |  |
| 复制工程 | copyProject 耗时：1.369 min | copyProject 耗时：653.9 ms |  |

# 欢迎交流

当然可能还有其他问题，欢迎交流;

也会有分析错误的地方，欢迎批评指正;

潘洪亮 QQ:28797575