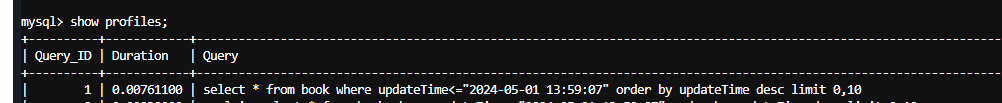
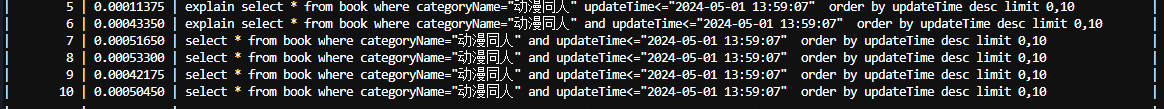
能通过这个指令，看到MySQL的“**线程池最大线程数量**”🡪但目前没找到用它进行计算的正确方式

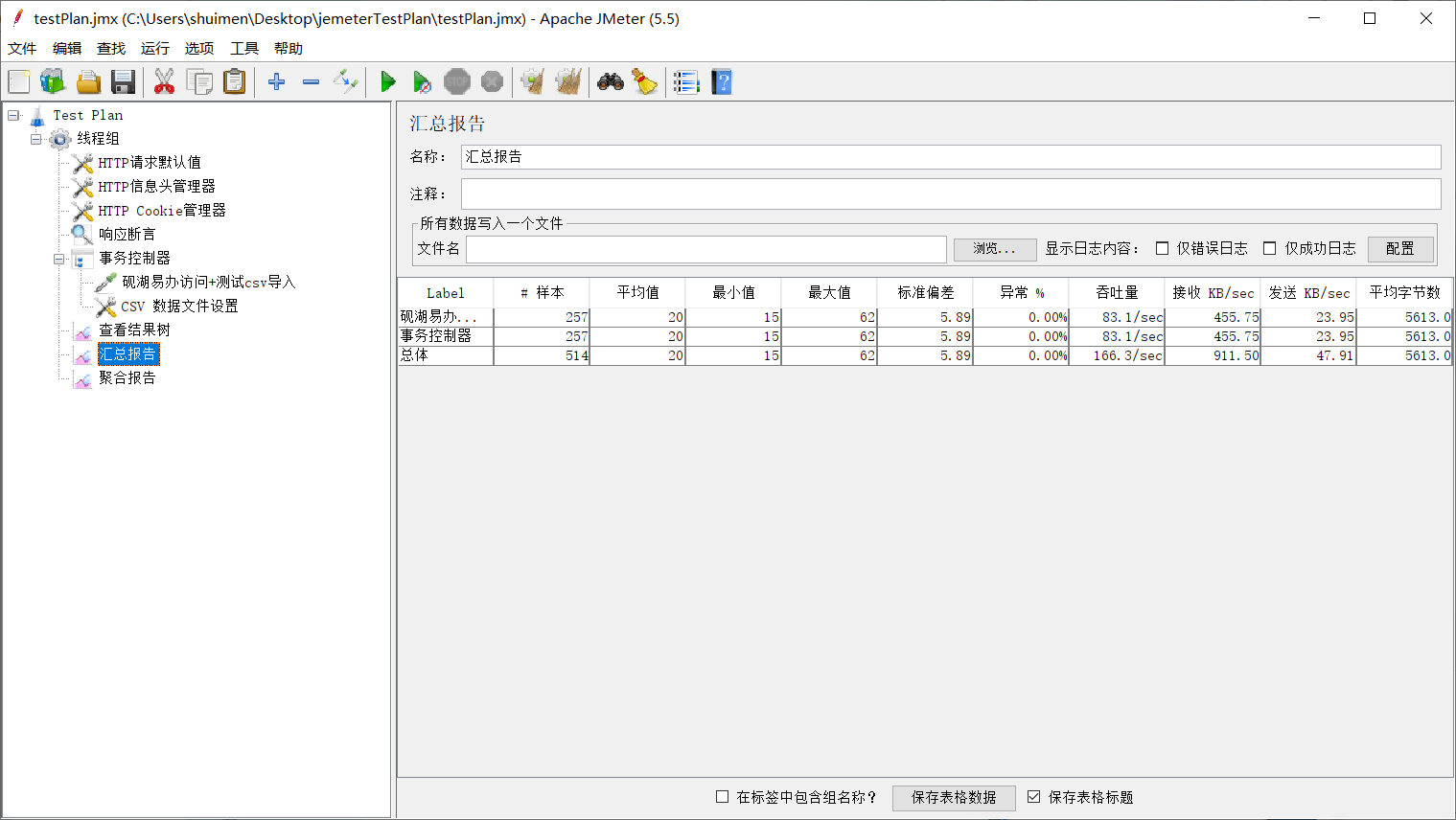




可以**看见用索引优化排序的情况，数量快了一个量级，76ms和5ms的对比**；

目前是**对updateTime设置了索引**【因为有些小说的lastChapterUpdateTime为null，不好弄】

**3s内处理257个请求**



Qps就是吞吐量，大概为**83.1左右**

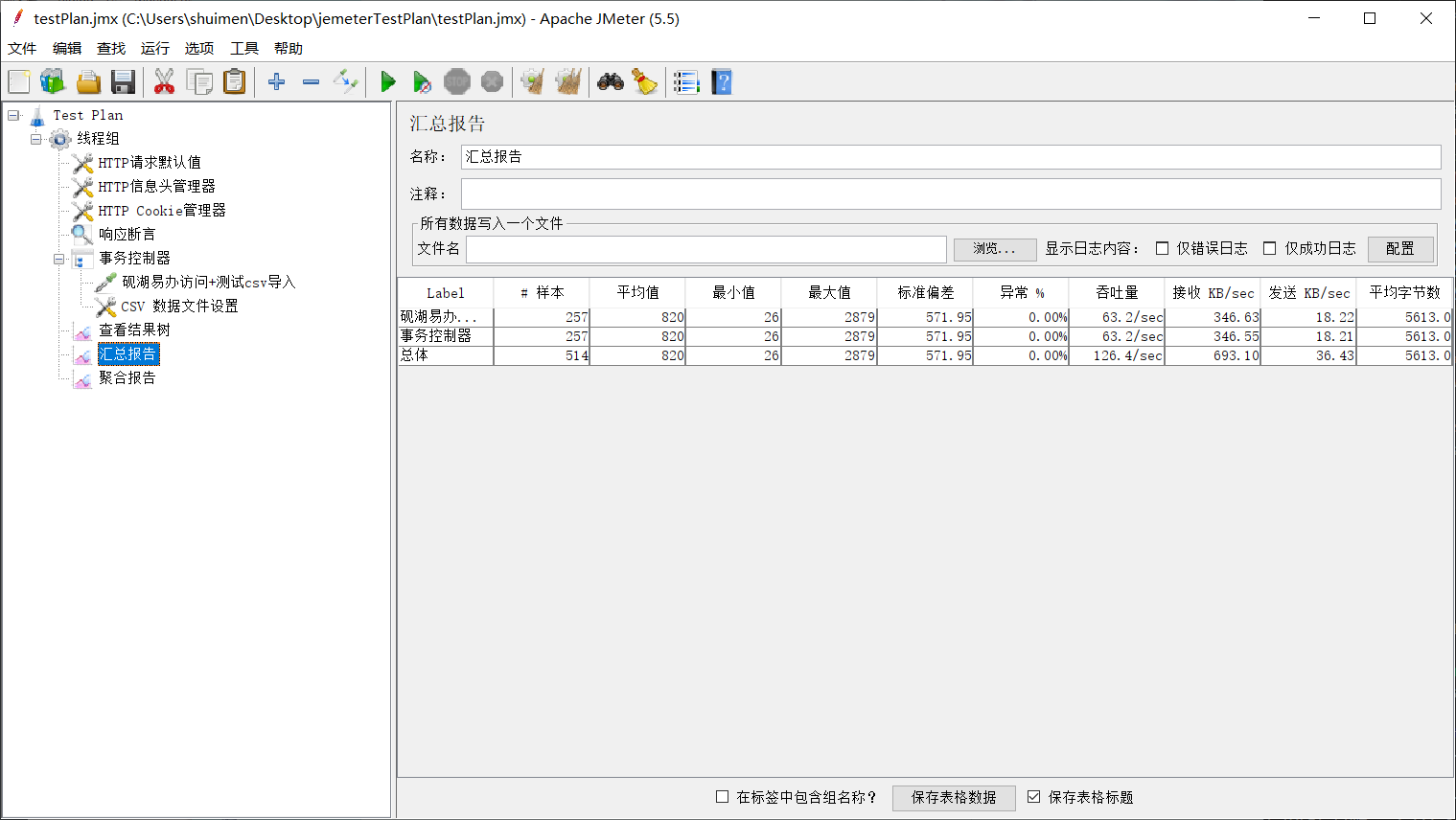
中途发现“用不用update索引时间上都差不多，存在问题”🡪**排查发现sql语句没进行order by排序🡪只能认为是“数据量不够多”，索引不能有效减少“io读取不常访问的数据页”**

**因此这里主要还是“优化了排序效率”**

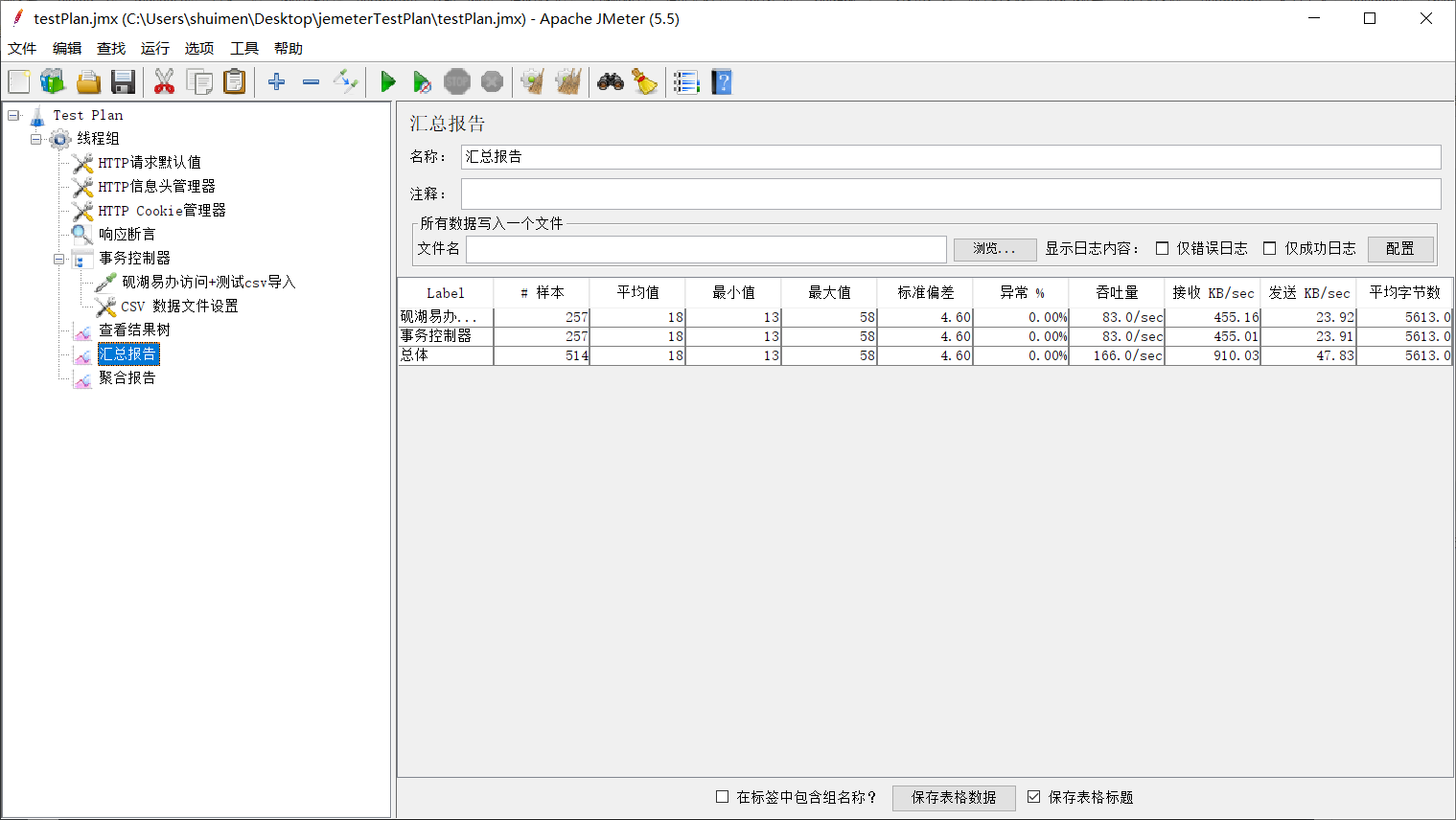
**看了看，目前缓存池大小为128mb【阿里云服务器配置】，如果16kb为一页，那么就会有8192个数据页能常驻？ 一个行数据大小为2336字节（主要book表的desc占用字节较多），一个数据页能保存完整的6个行数据🡪因此内存中完全能保留6x8192个行数据，也就是4w多的行数据，比你现在的小说表数据多得多🡪至少要超过4w的行数据，才会说考虑到“缓存命中率”问题**

**show GLOBAL VARIABLES LIKE '%innodb\_buffer\_pool\_size%'; 查看mysql的buffer pool数量**

**不用update字段作为索引的情况（这种情况应该进行的是全表where扫描）**，**qps为63.2**左右【注意，是基于阿里云服务器的配置】

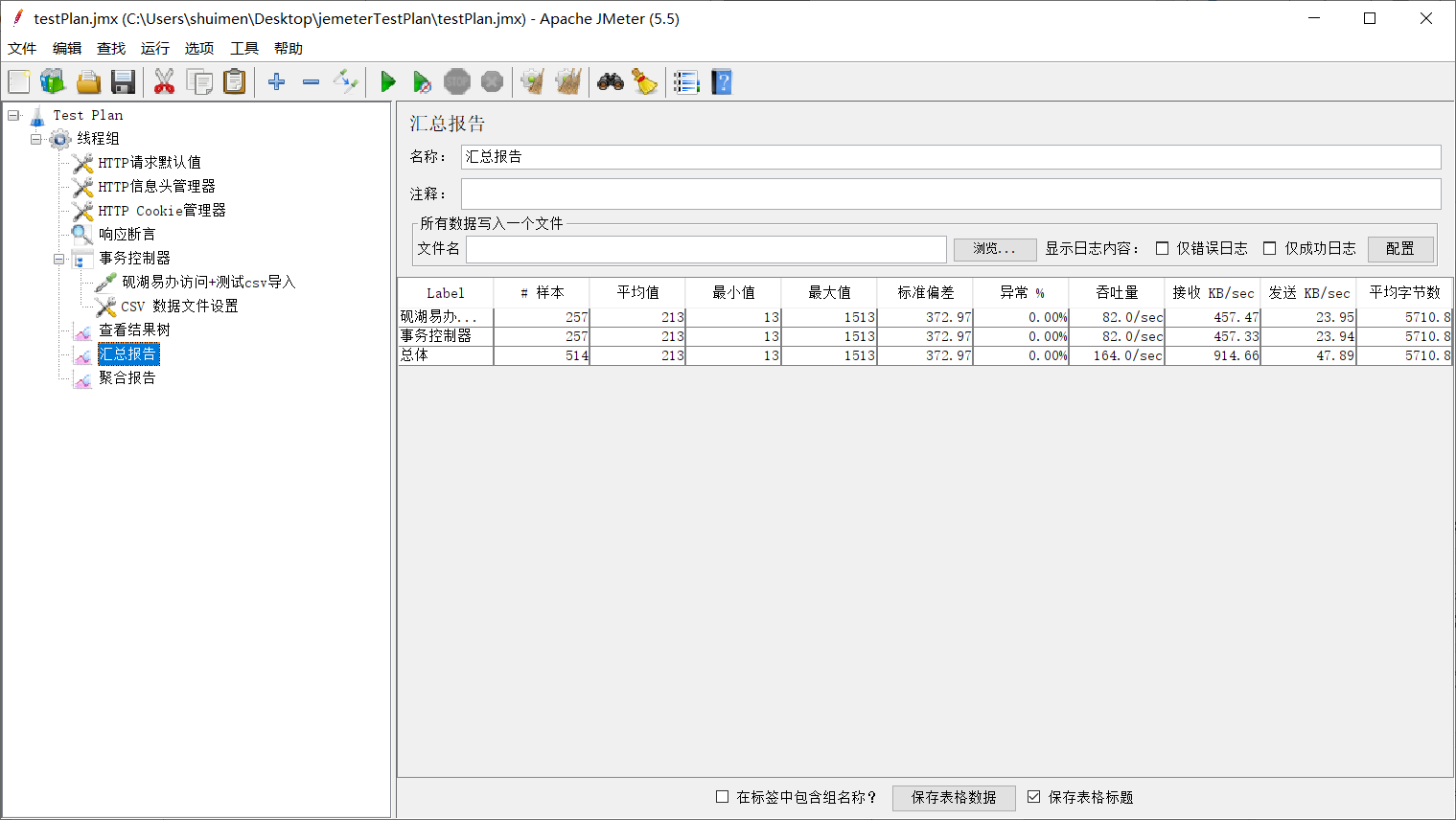


用**categoryName+updateTime做联合索引**的情况：



**Qps大概为83左右**，和只对updateTime建立索引时差不多

**下面是“需要收集点赞数量”的情况🡪2s257个**：



直接获取所有like数据，再统计数量的情况，**qps为83.0**【其实没多多少】

一个个count语句执行的情况：

