quarrier工程，为二进制建索引，用于利用指纹快速检索文档

创建原因：

为听歌识曲所写的指纹检索服务，由于为了顾及几百万的音频，以及在大量音频的情况下性能问题，对内存的访问次数几利用率到了变态的地步，笨重又不容易扩展。此时如果有一些其他的服务需要用音频指纹来识别音频，而这些服务的音频数量比较小，如果将这些音频加到虾米的指纹库中，或多或少会相互影响；如果用当前的指纹服务为这些服务搭建一个新的服务，又太笨重，虾米听歌识趣为了速度，对音频指纹的键值用完美hash进行映射，这样仅仅倒排索引表就要占用4G多的内存，更何况为了更高效的使用内存，虾米的音频指纹服务用c++编写，服务的搭建和扩容都不方便。综上种种情况，希望可以用java写一个针对小量指纹库的索引服务，方便以后集团中类似的需求。

模块设计：

1. 建库模块



1. 建库主线程或者实时添加文档的用户线程创建doc对象添加到creator中，creator维护一个线程安全的doc list，当文档数达到阈值时，创建一个createTask并清空列表，添加到调度中心（任务调度中心就是我之前讲的线程池）。
2. 工作线程获取createTask并执行，为这批doc创建倒排索引和正排索引，并创建segment对象持有倒排索引和正排索引。创建完后将segment添加到Merger对象中。
3. Merger对象接收segment对象，并保存在一个多线程安全队列中，当segment数量达到一个阈值时，创建一个mergeTask并清空列表，然后将mergeTask添加到任务调度中心。
4. 工作线程获取mergeTask，将这批segment合并成一个更大的segment，通过判断设置的合并深度或者segment大小，或者将segment添加到merge对象中，或者将segment添加到writer对象中
5. Writer对象接受segment对象，也维护一个多线程安全的segment列表，如果segment数量大于阈值，创建writerTask，清空列表，并将writerTask添加到任务调度中心。
6. 工作线程获取writertask，对这一批segment执行写磁盘的操作。
7. 磁盘合并segment模块（暂时不考虑，若考虑时，请思考下面几个问题）
8. Writer在写完每个segment时需要向一个固定文件.segid写入这个segment的id（这里要添加读写锁）
9. Load模块和diskMerge模块都要去读取.segid文件，以得知当前有哪些segment。
10. DiskMerge只能唯一执行，如果某个线程在执行diskMerge的操作，其他线程再执行时直接返回，这里应该设置一个状态来标识diskMerge是否在执行。
11. DiskMerge完成后，会将自己合并的segmentid写入.segid文件，并将合并前的文件删除
12. 合并后通知load，从新加载新的索引文件
13. 检索模块

检索模块重要的两个问题：1. 所有的数据必须读入内存，挑选合适的结构，尽可能的节省内存。2. 每个音频的指纹包含大量的子指纹，虾米曲库中平均每个指纹80K，就是2万个子指纹，这样10000首音频就有2亿个子指纹，假设有1半是重复的，这样建库的key值就有1亿个…….，不禁要打个冷战，满二叉树找到一个子指纹最多要30次比较，红黑树的深度大概是2倍，需要60次比较。所以要挑选一个合适的倒排检索结构（c++版用的是完美hash，需要占用大量的内存）

1. Load模块将磁盘中所有的segment 读入内存，包括倒排索引表，到排项列表，正排索引表，正排项列表。
2. 单线程查询模式：Search模块根据用户输入的查询请求后，逐个查询所有segment，查到符合条件的第一个结果后返回。
3. 多线程查询模式：为了加快单个查询的检索速度，设计一个多线程检索模式，Search模块接收到用户输入的查询后，为每个segment创建一个searchtask，放入调度中心模块。Search模块接收到任何一个线程的结果后，设置所有task的一个volatile变量，使其他线程结束搜索任务。



1. 实时检索模块：（暂时不考虑）