分片的内部原理

- 什么是ES的分片

  ES中的最小工作单元，是一个Lucene的index

- 一些问题

  为什么ES的搜索是近实时的(1秒后被搜索到)

  ES如何保证在断电时数据也不会丢失

  为什么删除文档，并不会立即释放空间

倒排索引不可变性

- 倒排索引采用Immutable Design，一旦生成，不可更改
- 不可变性，带来了的好处：

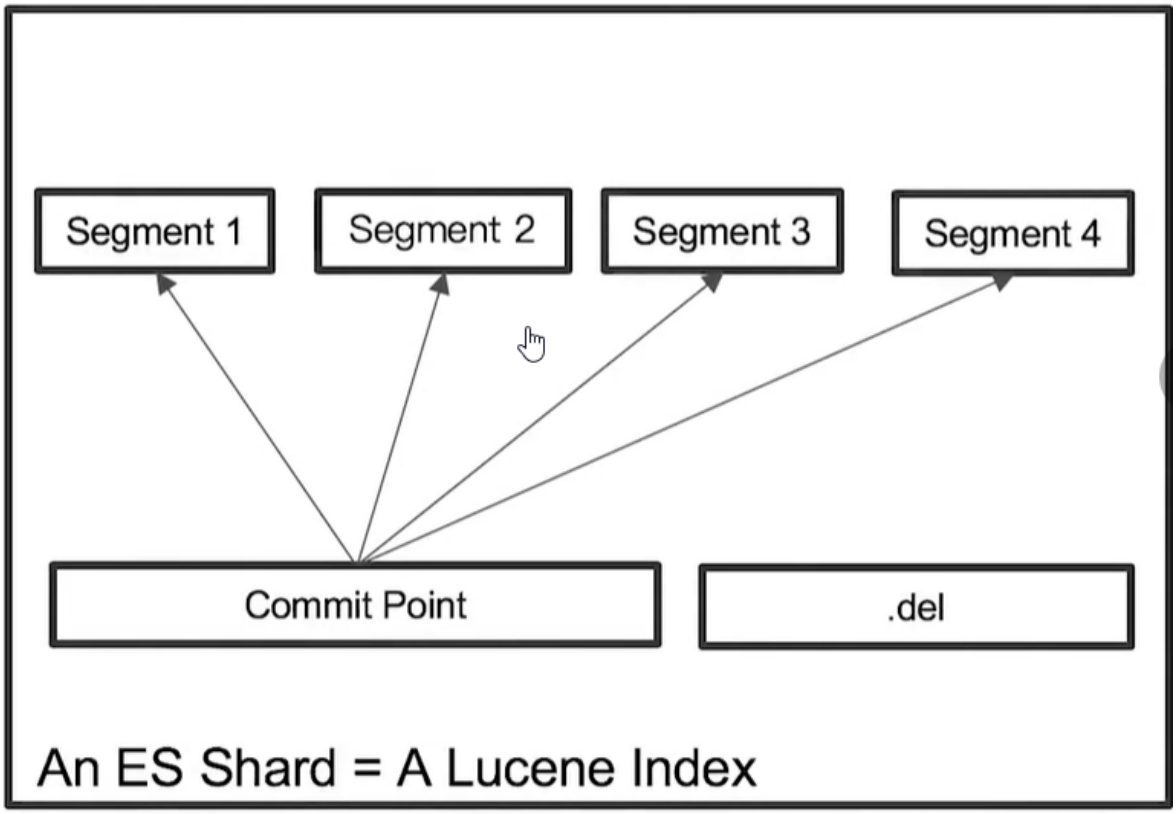
  （1）无需考虑并发写文件的问题，避免锁机制带来的性能问题

  （2）一旦读入内核的文件系统，便留在那里。只要文件系统内存有足够的空间，大部分请求就会直接请求到内存，不会命中磁盘，提升了很大的性能

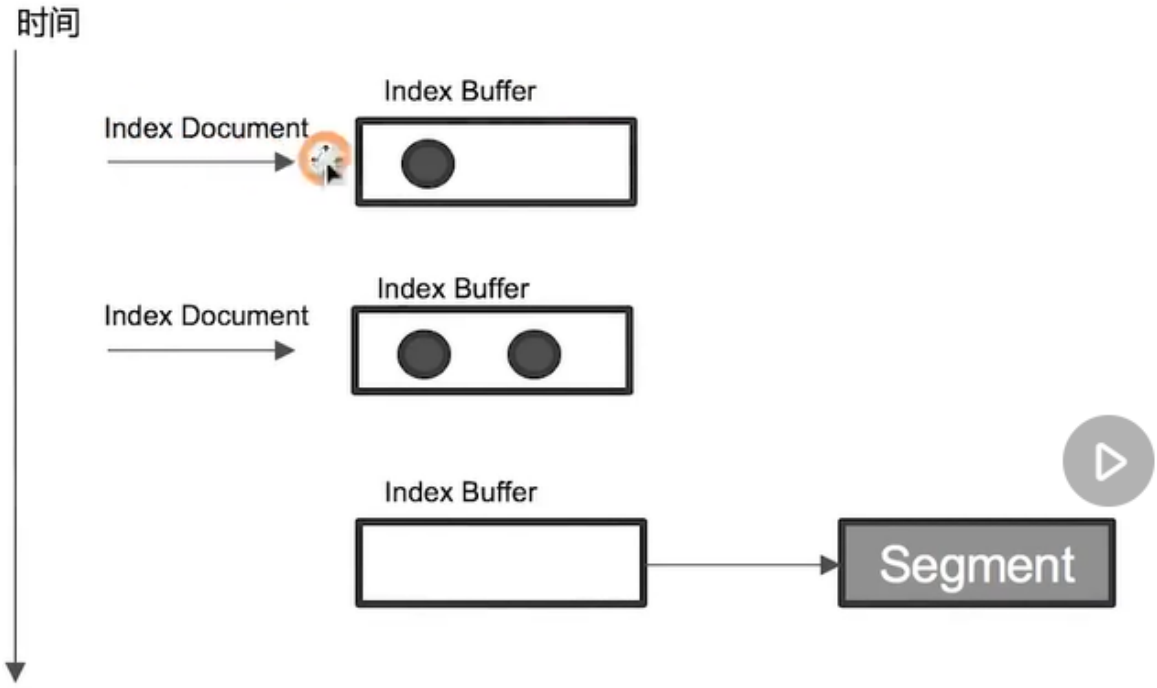
  （3）缓存容易生成和维护，数据可以被压缩
- 带来的挑战：如果让一个新的文档可以被搜索，需要重建整个索引。

Lucene Index

- 在Lucene中，单个倒排索引文件被称为Segment。Segment是自包含的，不可变更的。多个Segment汇总在一起，称为Lucene的index，其对应的就是ES中Shard。
- 当有新文档写入时，会生产新Segment，查询时会同时查询所有Segments，并对结果汇总。Lucene中有一个文件，用来记录所有Segments信息，叫做Commit Point。
- 删除的文档信息，保存在".del"文件中

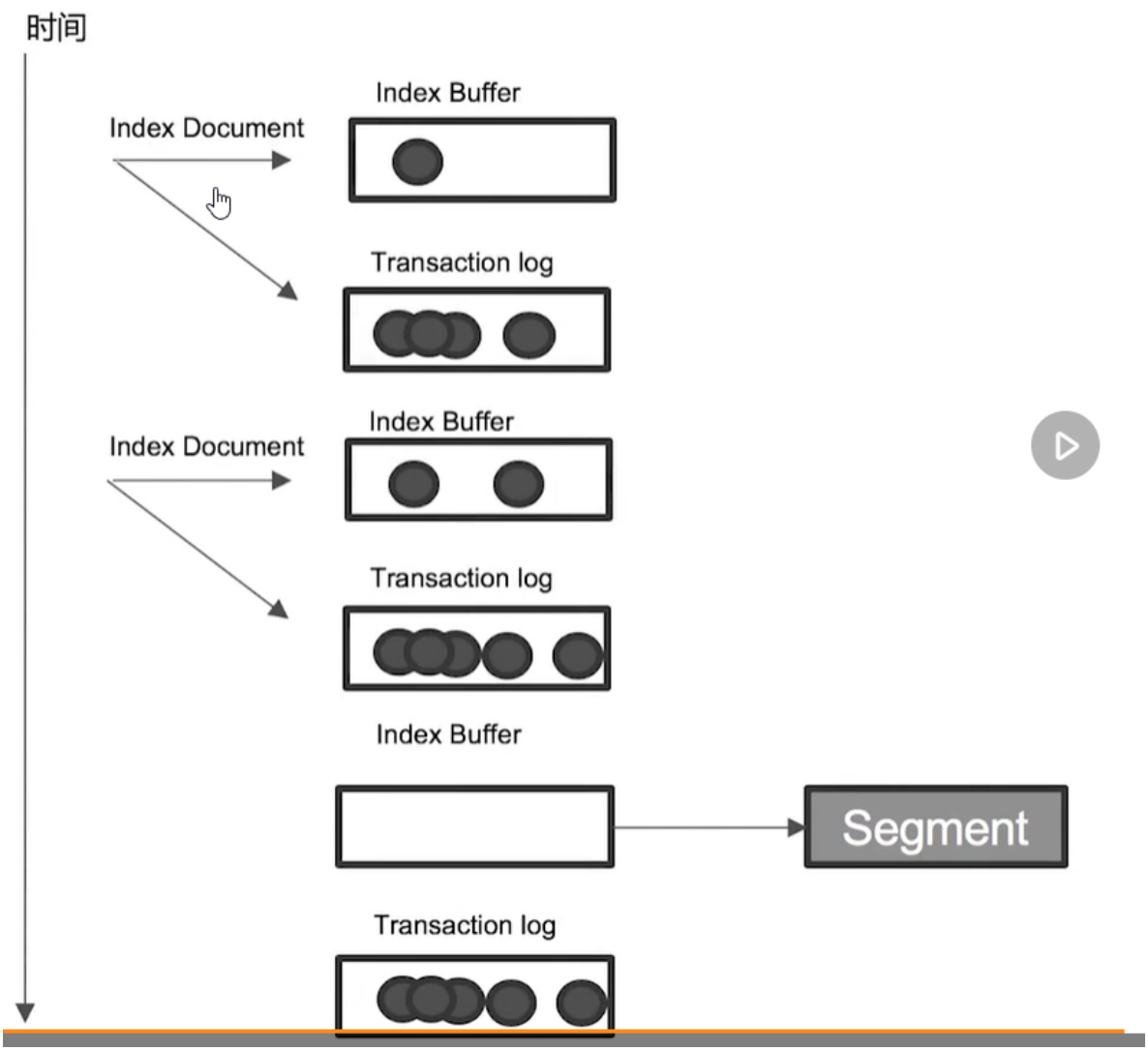


Refresh



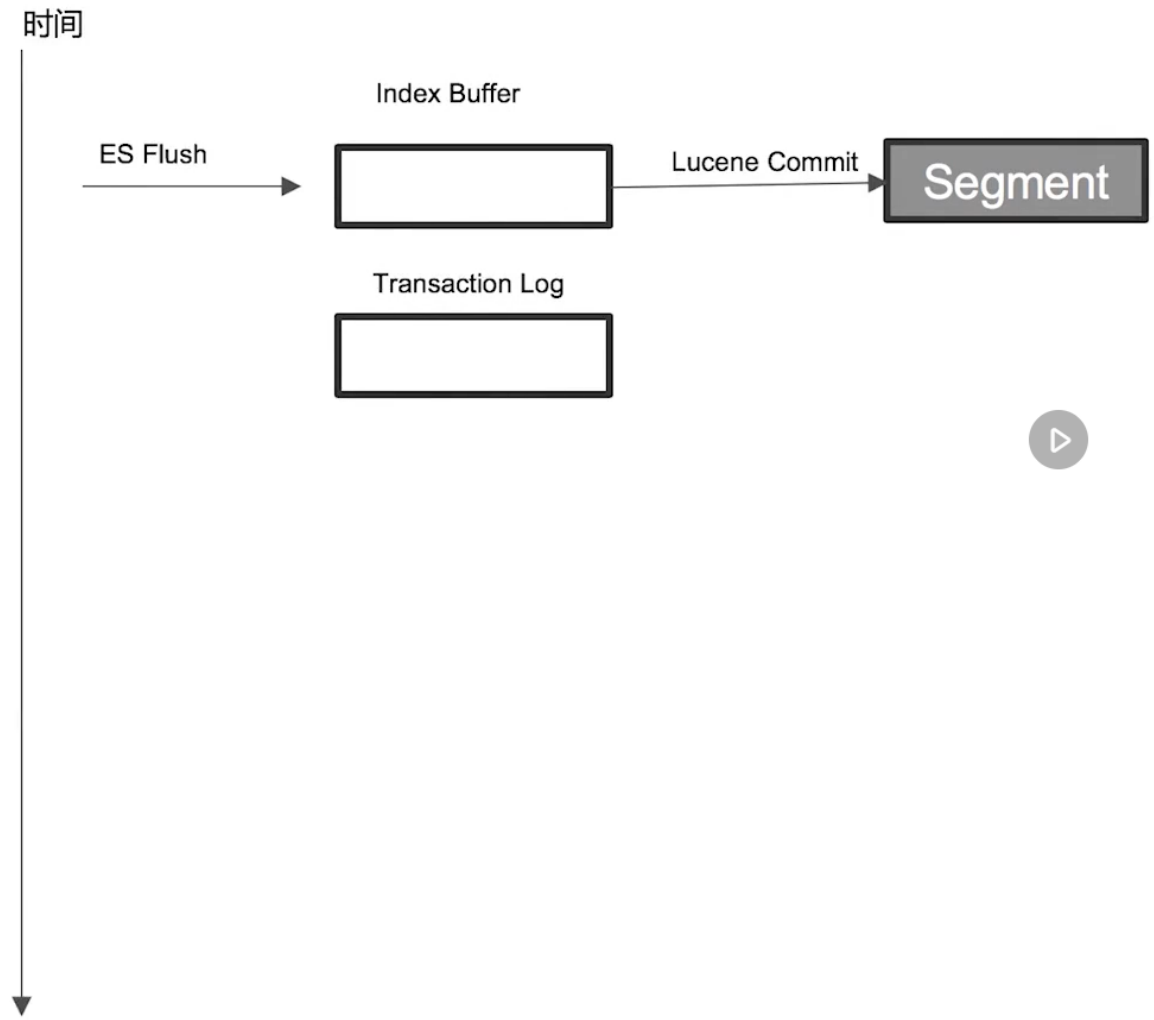
- 将Index buffer写入Segment的过程叫Refresh。**Refresh不执行fsync操作。**
- Refresh频率：默认是1秒发生一次，可通过index.refresh_interval配置。Refresh后，数据就可以被搜索到了。**这解释了为啥ES被称为近实时搜索。**
- 如果系统有大量的数据写入，那就会产生很多的Segment。
- Index Buffer 被沾满时，会触发Refresh，默认值是JVM的10%。

Transaction Log



- Segment 写入磁盘的过程相对耗时，借助文件缓存系统，Refresh时，先将Segment写入缓存以开放查询。
- 为了保证数据不会丢失。所以在Index文档时，同时会写Transaction Log，高版本开始时，Transaction Log默认落盘。每个分配有一个Transaction Log。
- 在ES Refresh时，Index Buffer被清空，Transaction Log不会被清。

Flush



ES Flush & Lucene Commit：

- 调用Refresh，Index Buffer清空并且Refresh
- 调用fsync，将缓存中的Segments写入磁盘
- 清空(删除)Transaction Log
- 默认30分钟调用一次
- Transaction Log满（默认512M）

Merge

- Segment 很多，需要被定期合并

  减少Segment，删除已经删除的文件
- ES和Lucene会自动进行Merge操作

  POST my_index/_forcemerge