

现有流程的问题，每次都必须等待 fsync 将 segment 刷入磁盘，才能将 segment 打开供 search 使用，这样的话，从一个 document 写入，当被它搜到，可能超过一分钟！！！这就不是近实时的搜索了，主要的瓶颈在于 fsync 实际发生磁盘IO 写入数据磁盘是很耗时的

写入流程改进如下：(数据只要被写入 os cache 就直接可以搜索了，而不是写入磁盘才可以被搜索，这样写入 cache 大概需要 1秒，所以es 实现近实时)

1. 数据写入 buffer
2. 每隔一定时间，buffer 中的数据被写入 segment 文件，但是先写入 os cache
3. 只有 segment 写入 os cache，那就直接打开供 search 使用，不立即commit

数据写入 os cache，并被打开供搜索的过程，叫做 **refresh**，默认是每隔 1 秒 refresh 一次，也就是说，每隔一秒就将 buffer 中的数据写入一个新的 index segment file，先写入 os cache。所以，es 是近实时的，数据写入到可以被搜索，默认是 1秒

POST /my_index/_refresh，可以手动 refresh，一般不需要手动执行，没必要，让es 自己刷新就可以了

比如说，我们现在的时效性要求比较低，只要求一条数据写入 es，一分钟以后才让我们搜索到就可以了，那么就可以调整 refresh interval

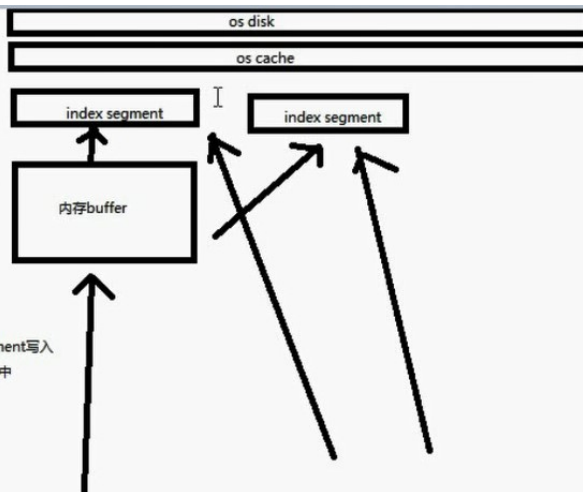
```
1 PUT /my_index
2 {
3     "settings": {
4         "refresh_interval": "30s"
5     }
6 }
```

这个时候，改进流程的地方到了，我们不会等待fsync将os cache中的数据刷如os disk，才将index segment打开供search使用。而是index segment数据一到os cache中，就立即打开，供search使用

index segment数据先写入os cache中

每到一定时间，buffer中的数据别写入一个新的index segment中

一条document写入内存buffer中



每个秒，buffer被刷新到一个新的index segment中，所以每秒都会产生一个新的index segment file