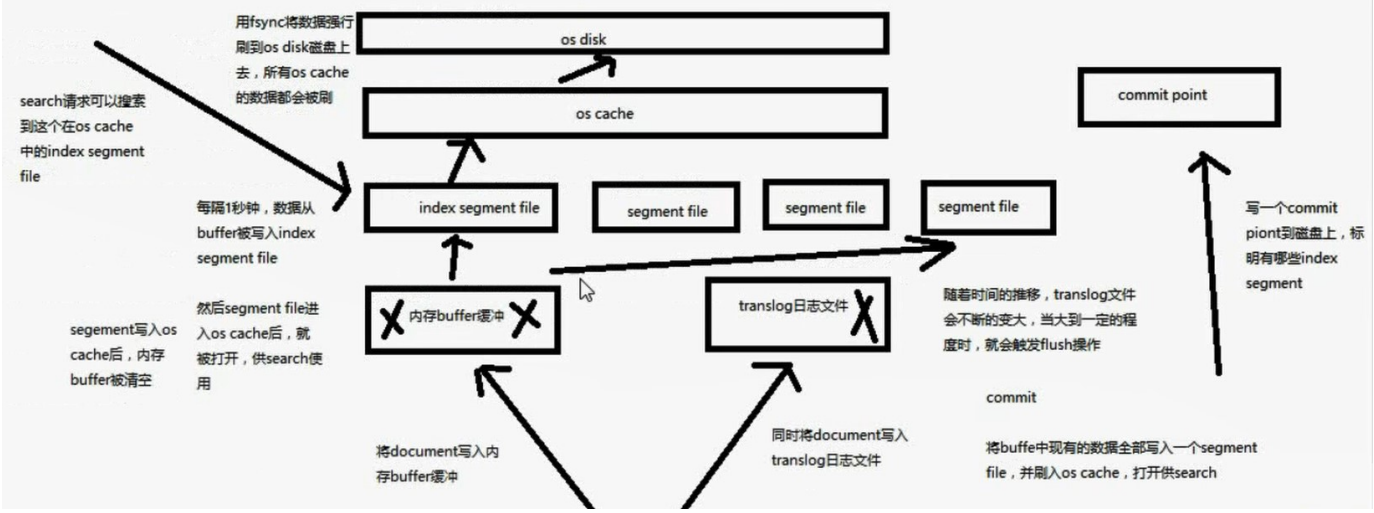
再次优化写入流程

1. 数据写入 buffer 缓冲和 translog 日志文件
2. 每隔一秒，buffer 中的数据被写入到新的 segment file，并进入 os cache，此时segment被打开并供 search 使用
3. buffer 被清空
4. 重复 1~3 新的 segment 不断增加，buffer 不断被清空，而 translog 中的数据不断累加
5. 当 translog 长度达到一定程度的时候，commit 操作发生
   5-1. buffer 中的所有数据写入一个新的 segment，并写入 os cache，并打开供使用
   5-2. buffer 被清空
   5-3. 一个commit point 被写入磁盘，标明了所有的 index segment
   5-4. filesystem cache 中的所有 index segment file 缓存数据，被fsync 强行刷到磁盘上
   5-5. 现有的 translog 被清空，创建一个新的 translog



基于 translog 和 commit point，如何进行数据恢复

fsync + 清空translog，就是 flush，默认每隔30分钟 flush 一次，或者当translog 过大的时候，也会 flush

```
POST  /my_index/_flush            //手动flush 的命令
```

translog，每隔 5秒被 fsync 一次到磁盘上，在一次增删改操作之后，当 fsync 在 primary shard 和 replica shard 都成功之后，那次的增删改操作才会成功

但是这种在一次增删改时强行 fsync translog 可能会导致部分操作比较耗时，也可以允许部分数据丢失，设置异步 fsync translog

```
PUT /my_index/_settings
{
    "index.translog.durability":"async",
    "index.translog.sync_interval":"5s"
}
```