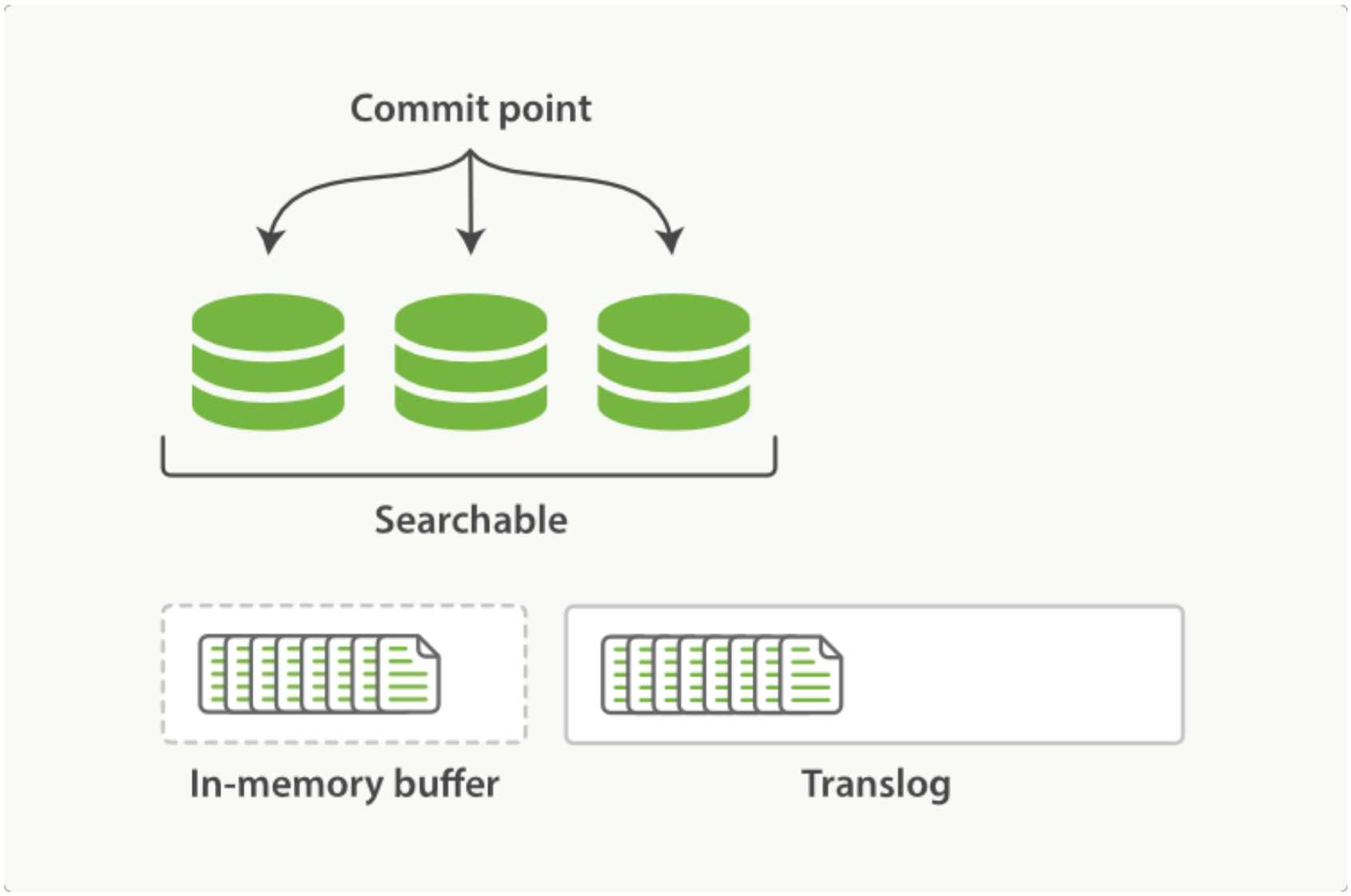
Elasticsearch如何保证数据不丢失?

上篇文章提到过，在elasticsearch和磁盘之间还有一层cache也就是filesystem cache，大部分新增或者修改，删除的数据都在这层cache中，如果没有flush操作，那么就不能100%保证系统的数据不会丢失，比如突然断电或者机器宕机了，但实际情况是es中默认是30分钟才flush一次磁盘，这么长的时间内，如果发生不可控的故障，那么是不是必定会丢失数据呢?

很显然es的设计者早就考虑了这个问题，在两次full commit操作（flush）之间，如果发生故障也不能丢失数据，那么es是如何做到的呢?

在es里面引入了transaction log（简称translog），这个log的作用就是每条数据的任何操作都会被记录到该log中，非常像Hadoop里面的edits log和hbase里面的WAL log，如下图:



transaction log的工作流程如下:

（1）当一个文档被索引时，它会被添加到内存buffer里面同时也会在translog里面追加

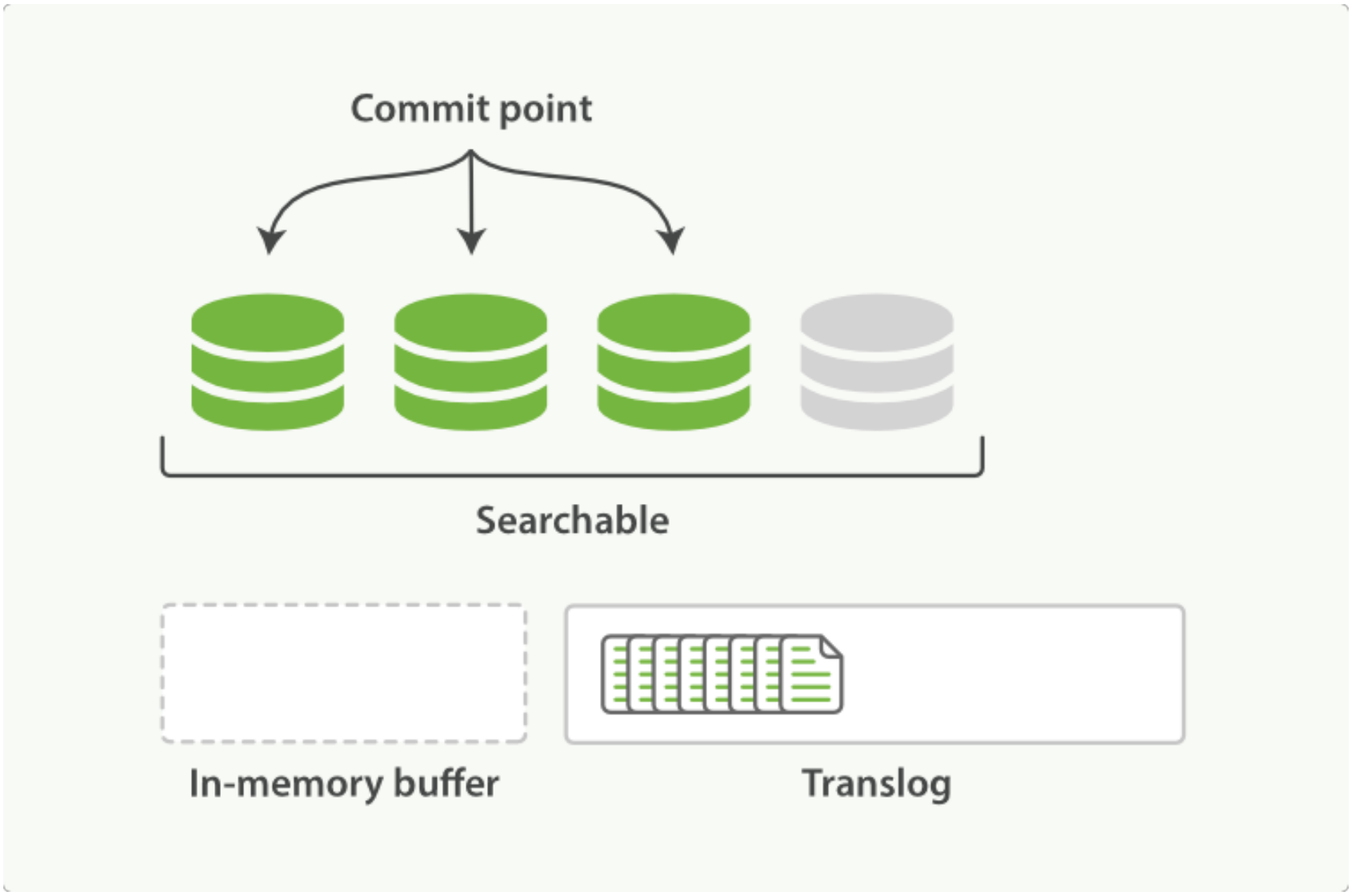
（2）当每个shard每秒执行一次refresh操作完毕后，内存buffer会被清空但translog不会。

过程如下：

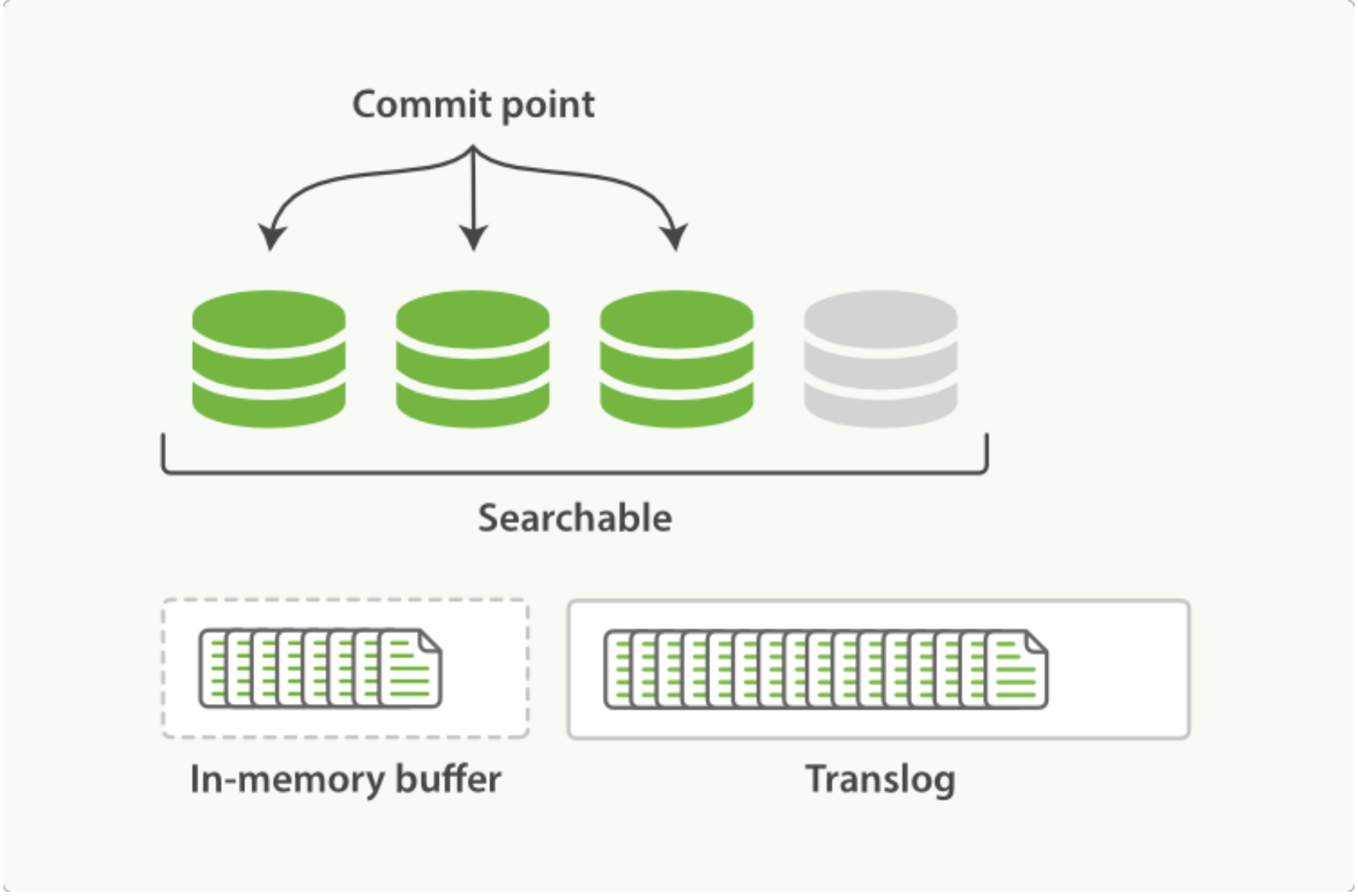
2.1 当refresh动作执行完毕后，内存buffer里面的数据会被写入到一个segment里面，这个还在**cache**中，并没有执行**flush**命令

2.2 新生成的**segment**在**cache**中，会被打开，这个时候就可以搜索新加的数据

上面过程如下图：



（3）随着更多的document添加，内存buffer区会不断的refresh，然后clear，但translog数量却越增越多，如下图：



（4）当达到默认的30分钟时候，translog也会变得非常大，这个时候index要执行一次flush操作，同时会生成一个新的translog文件，并且要执行full commit操作，流程如下：

4.1 内存buffer里的所有document会被生成一个新的segment

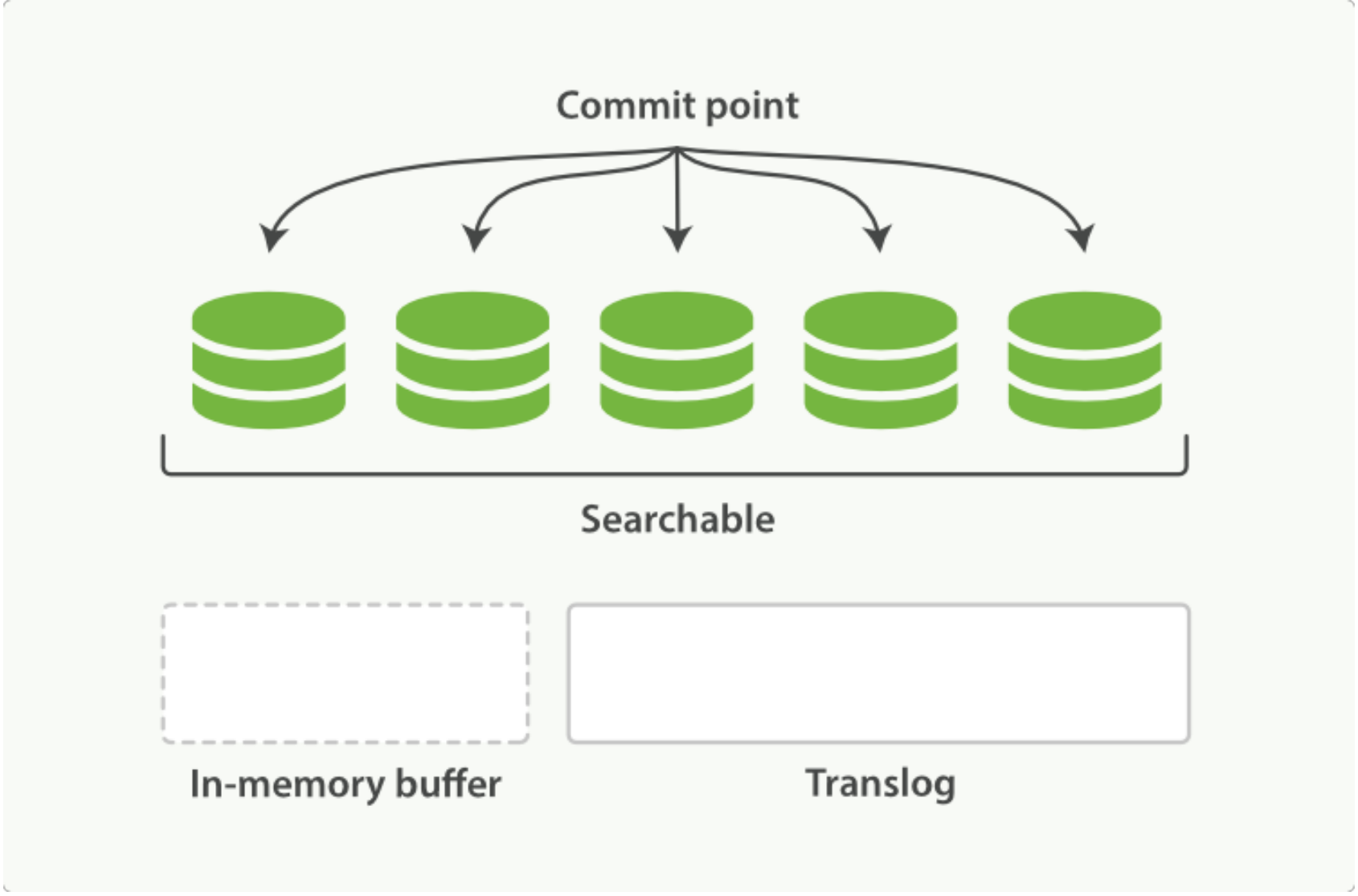
4.2 然后segment被refresh到系统cache后，内存buffer会被清空

4.3 接着commit point会被写入到磁盘上

4.4 filesystem cache会被flush到磁盘上通过fsync操作

4.5 最后旧的translog会被删除，并会生成一个新的translog

如下图：



tanslog的作用就是给所有还没有flush到硬盘上的数据提供持久化记录，当es重启时，它首先会根据上一次停止时的commit point文件把所有已知的segments文件给恢复出来，然后再通过translog文件把上一次commit point之后的所有索引变化包括添加，删除，更新等操作给重放出来。

除此之外tanslog文件还用于提供一个近实时的CURD操作，当我们通过id读取，更新或者删除document时，es在从相关的segments里面查询document之前，es会首先从translog里面获取最近的变化，这样就意味着es总是近实时的优先访问最新版本的数据。

我们知道执行flush命令之后，所有系统cache中的数据会被同步到磁盘上并且会删除旧的translog然后生成新的translog，默认情况下es的shard会每隔30分钟自动执行一次flush命令，或者当translog变大超过一定的阈值后。

flush命令的api如下：

```
POST /blogs/_flush  //flush特定的index

POST /_flush?wait_for_ongoing//flush所有的index知道操作完成之后返回响应
```

flush命令基本不需要我们手动操作，但当我们要重启节点或者关闭索引时，

最好提前执行以下flush命令作为优化，因为es恢复索引或者重新打开索引时，它必须要先把translog里面的所有操作给恢复，所以也就是说translog越小，recovery恢复操作就越快。

我们知道了tangslog的目的是确保操作记录不丢失，那么问题就来了，tangslog有多可靠?

默认情况下，translog会每隔5秒或者在一个写请求（index，delete，update，bulk）完成之后执行一次fsync操作，这个进程会在所有的主shard和副本shard上执行。 这个守护进程的操作在客户端是不会收到200 ok的请求。

在每个请求完成之后执行一次translog的fsync操作还是比较耗时的，虽然数据量可能比并不是很大。 默认的es的translog的配置如下:

```
"index.translog.durability": "request"
```

如果在一个大数据量的集群中数据并不是很重要，那么就可以设置成每隔5秒进行异步fsync操作translog，配置如下:

```
"index.translog.durability": "async",
"index.translog.sync_interval": "5s"
```

上面的配置可以在每个index中设置，并且随时都可以动态请求生效，所以如果我们的数据相对来说并不是很重要的时候，我们开启异步刷新translog这个操作，这样性能可能会更好，但坏的情况下可能会丢失5秒之内的数据，所以在设置之前要考虑清楚业务的重要性。

如果不知道怎么用，那么就用es默认的配置就行，在每次请求之后就执行translog的fsycn操作从而避免数据丢失。