



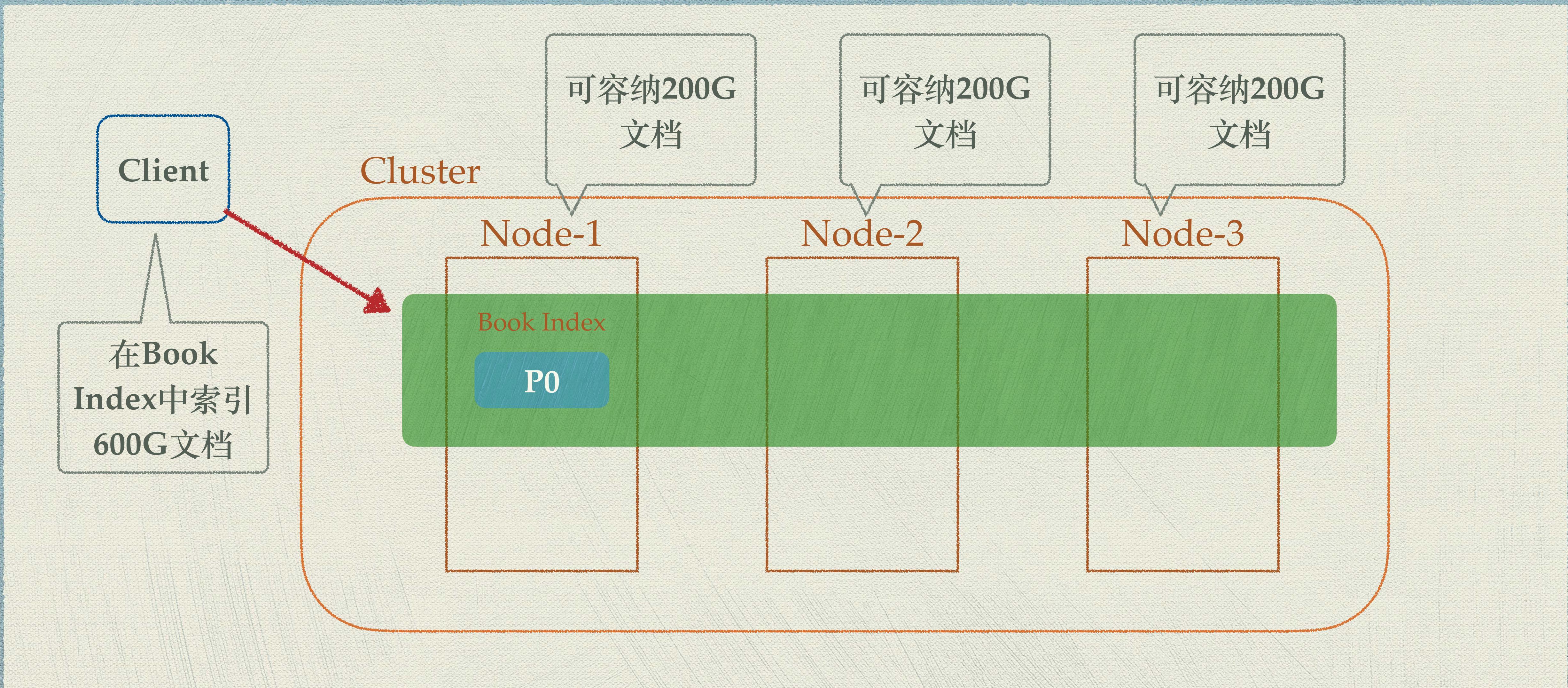
ElasticSearch

完全入门教程

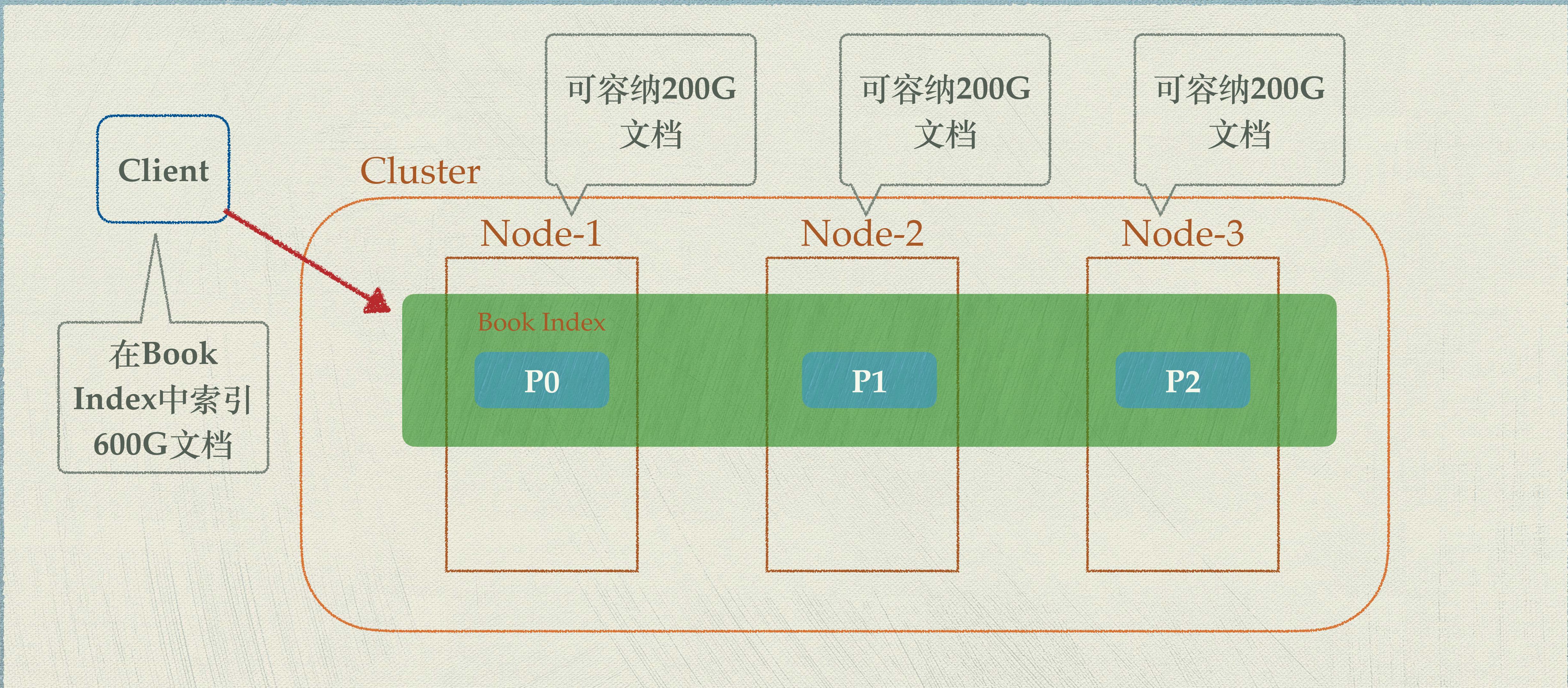
04. 分片 Sharding

波波微课

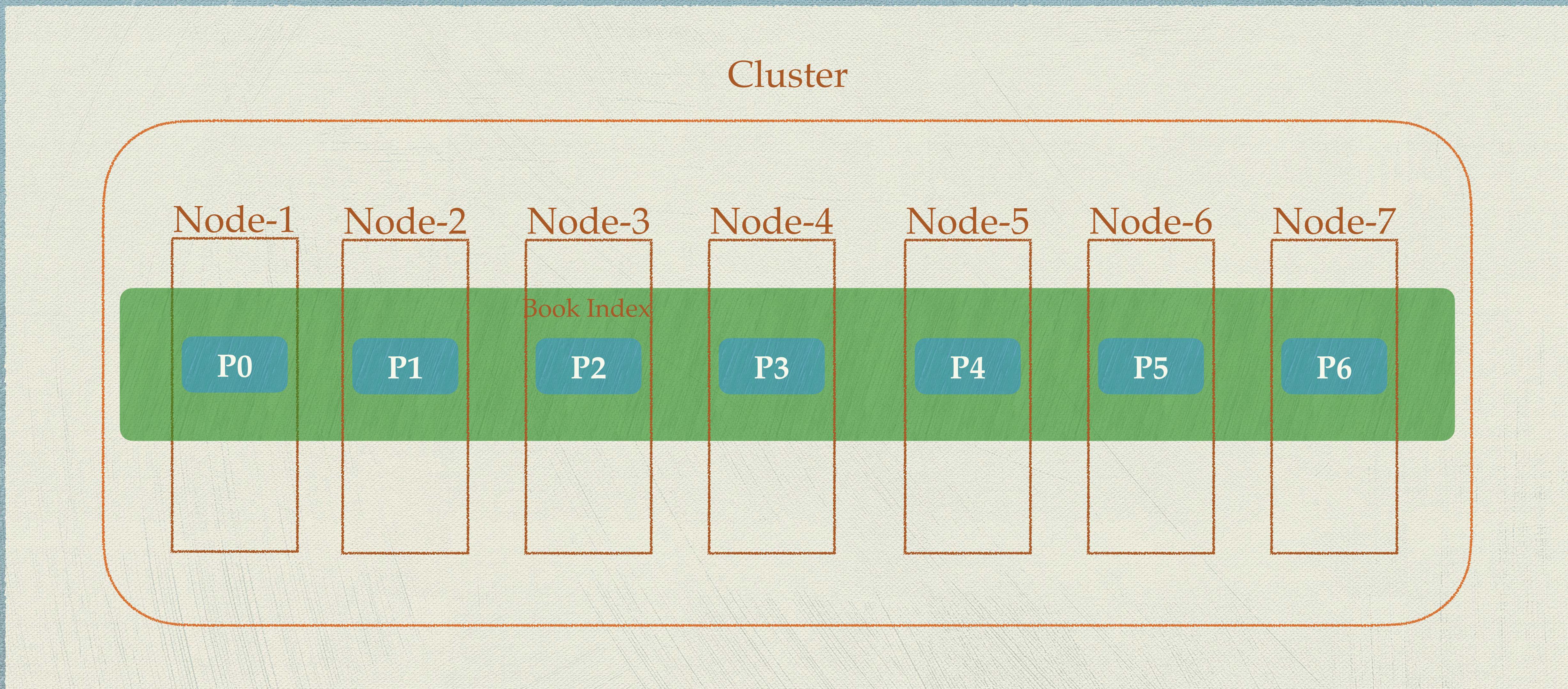
什么是分片Sharing?



什么是分片Sharing?

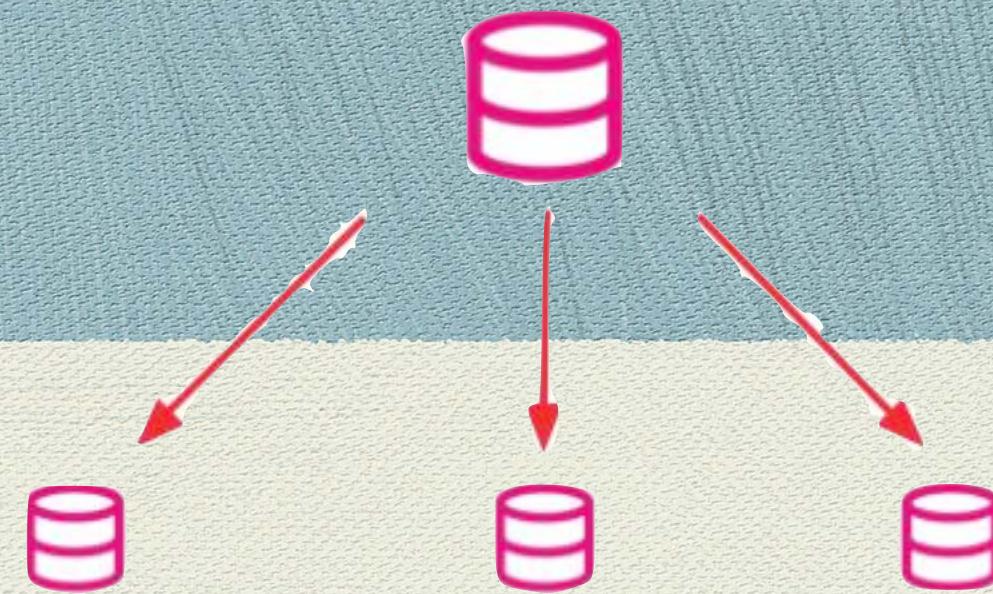


Sharing作用1 = 水平扩展



什么是Sharing?

- ◆ 将一个物理大索引拆分成若干个小索引
- ◆ 每一个小索引称为一个Shard
 - ◆ 一个Shard本质上是一个Lucene Index
- ◆ Index是逻辑概念，底层是若干个Shards，分布在若干节点上
- ◆ 一个Shard最多可存放20亿个文档



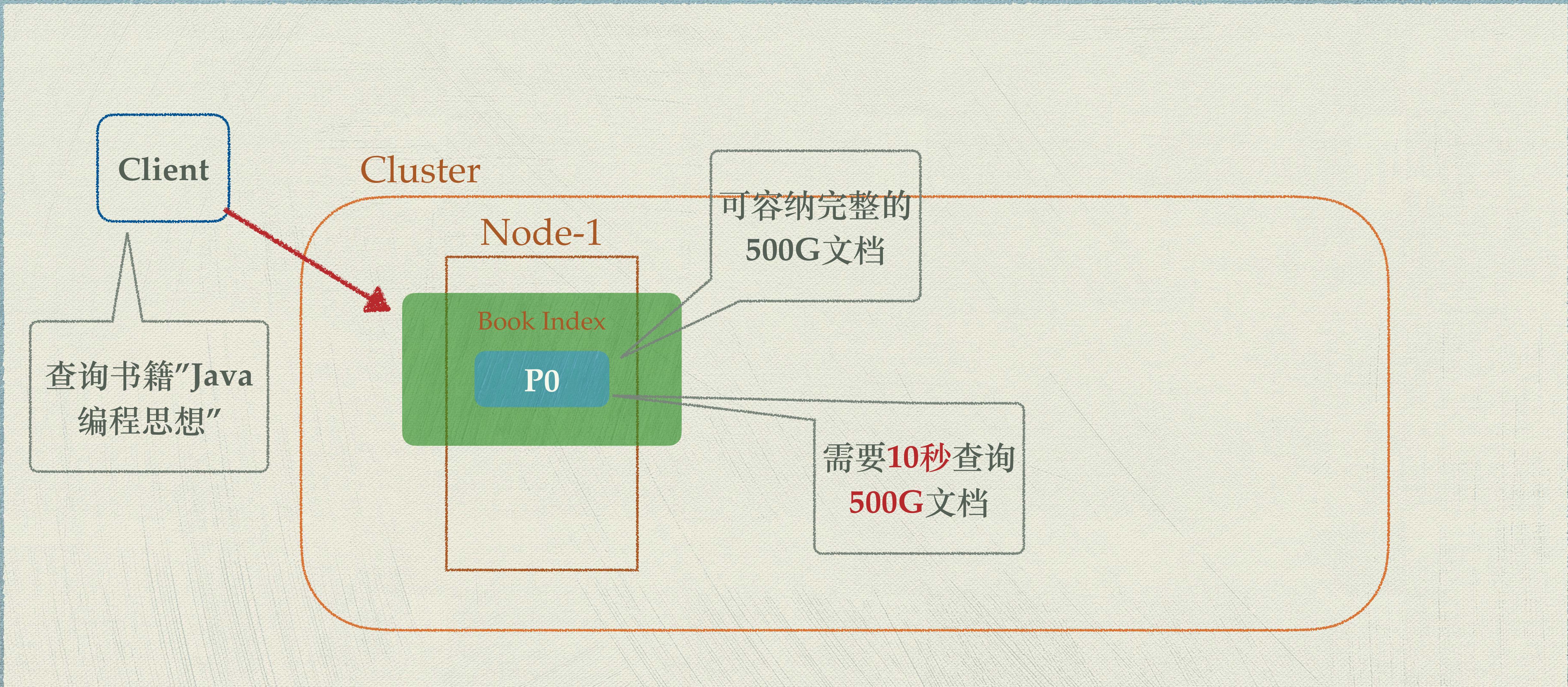
通过Kibana查看Shard

- ◆ GET /_cat/indices?v

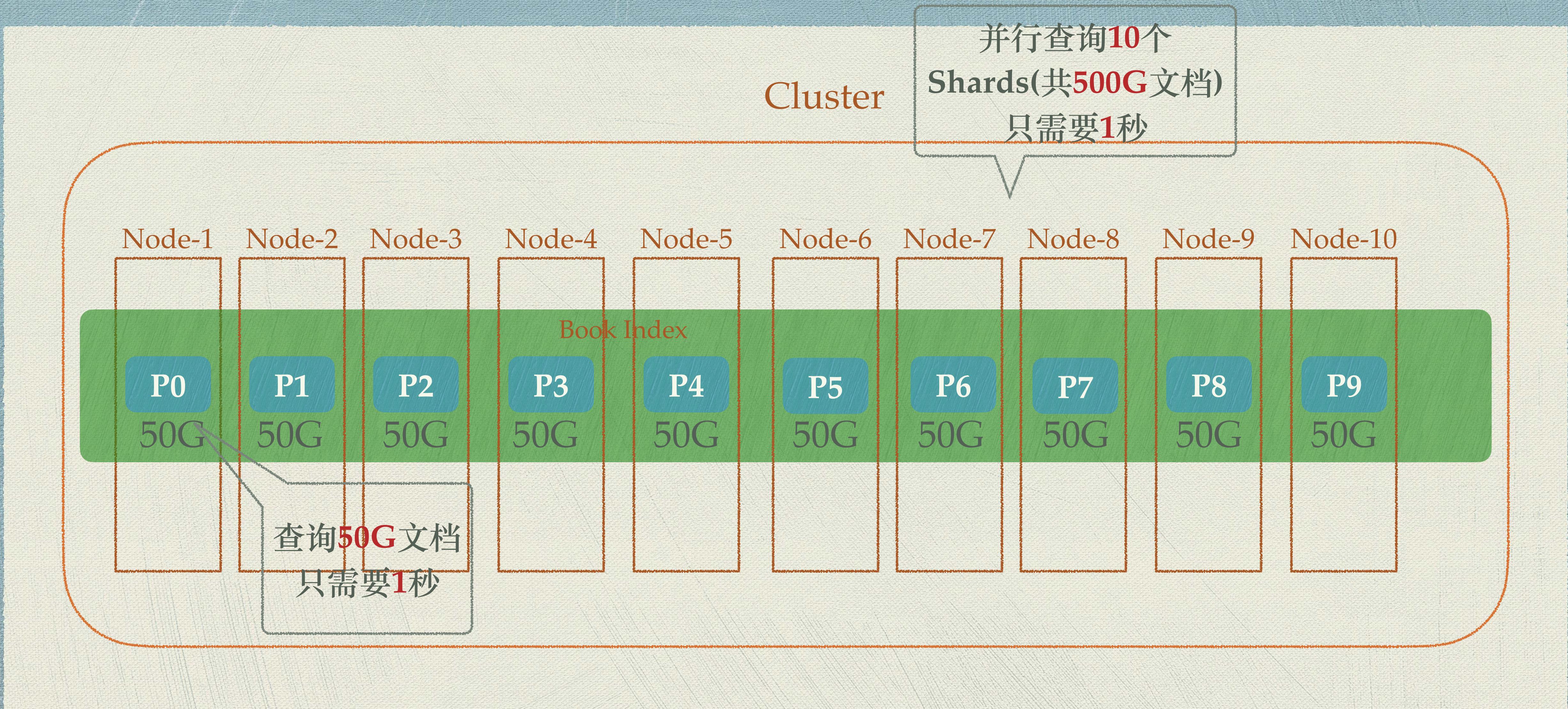
Sharding主要作用

- ◆ 支持数据量的水平扩展
- ◆ 将一个大物理索引拆分成若干的小索引，让每一个小索引都能存放在一个单独的节点上
- ◆ 提升查询性能

Sharding副作用 ~ 提升查询性能



Sharding副作用 ~ 提升查询性能



Sharding配置

- ◆ 一个Index缺省配置一个Shard(ES 7.x)
 - ◆ Shards并非越多越好，存储开销/管理复杂性
- ◆ 增加或减少Shards
 - ◆ 增加Shards -> Split API
 - ◆ 减少Shards -> Shrink API
- ◆ 配置多少个Shards合适?
 - ◆ 视情况定，节点数量，每个节点的容量，索引数量/大小，查询模式
 - ◆ 预期大索引 -> 适当多分配一些Shards(比方说从5个开始)
 - ◆ 中小索引 -> 从1/2个Shards开始
 - ◆ 监控+调整

本课小结



- ◆ Sharding分片可以将一个大的索引拆分成若干的小索引
 - ◆ 让一个索引的容量可以按需水平扩展
 - ◆ 让一个大的物理索引能够分别存放在各个存储容量有限的节点上
- ◆ Sharding也可以提升查询性能
- ◆ 一个Index缺省配置一个Shard(ES7.x)
- ◆ Sharding数量配置的最佳实践
 - ◆ 预期大Index，适当配置多一些Shards
 - ◆ 中小Index，从1/2个Shard开始