拉勾教育
互联网人实战大学

《31 讲带你搞懂 SkyWalking》

徐郡明　前搜狗资深技术专家、源码剖析系列畅销书作者

— 拉勾教育出品 —

拉勾教育
互联网人实战大学

第07讲：ElasticSearch基础入门

拉勾教育
L / A / G / O / U

ElasticSearch基础入门

拉勾教育

SkyWalking OAP 后端可以使用多种存储对数据进行持久化，如 MySQL、TiDB 等

默认使用 ElasticSearch 作为持久化存储

拉勾教育

ElasticSearch 基本概念

拉勾教育

Elasticsearch 是一个基于 Apache Lucene 的开源搜索引擎

Apache Lucene 被认为是迄今为止最先进、性能最好、功能最全的搜索引擎库

拉勾教育

ElasticSearch 基本概念

ElasticSearch 使用 Java 对 Apache Lucene 进行了封装

提供简单易用的 RESTful API

隐藏 Apache Lucene 的复杂性，降低了全文检索的编程门槛

拉勾教育
L / A / G / O / U

ElasticSearch 基本概念

拉勾教育

ElasticSearch	Databases
Indices	DB
Index	Table
Document	Row
Field	Column

老版本的 ElasticSearch 中，每个 Index 下可以建立多个 Type

从 ES 6.0 版本开始单个 Index 中只能有一个 Type

ES 7.0 版本以后将不建议使用 Type

ES 8.0 以后完全不支持 Type

拉勾教育 L / A / G / O / U

ElasticSearch 基本概念

拉勾教育

Document 是构建 Index 的基本单元

Document 以 JSON 格式表示，Field 则是这条 JSON 数据中的字段

```
{
  "_index": "order_index",
  "_type": "1",
  "_id": "kg72dW8BOCMUWGurlFiy",
  "_version": 1,
  "_score": 1,
  "_source": {
    "create_time": "2020-02-01
17:35:00",
    "creator": "xxxxx",
    "order_status":"NEW",
    "price": 100.00
  }
}
```

拉勾教育 L / A / G / O / U

ElasticSearch 基本概念

拉勾教育

Index 是具有某些类似特征的 Document 的集合

Index 与 Document 之间的关系类似于数据库中 Table 与 Row 之间的关系

在 Index 中可以存储任意数量的 Document

对 Document 的添加、删除、更新、搜索等操作，都需要明确的指定 Index 名称

拉勾教育

Index Template（模板）

拉勾教育

Index Template

一般包含 settings、mappings、index_patterns 、order、aliases

拉勾教育

Index Template（模板）

- index_patterns

  负责匹配 Index 名称，Index Template 只会应用到名称与之匹配的 Index 上

  而且 ElasticSearch 只会在 Index 创建的时候应用匹配的 Index Template

  后续修改 Index Template 时不会影响已有的 Index

  通过 index_patterns 匹配可以让多个 Index 重用一个 Index Template

拉勾教育
L / A / G / O / U

Index Template（模板）

拉勾教育

- settings

  主要用于设置 Index 中的一些相关配置信息，如分片数、副本数、refresh 间隔等信息

- mappings

  主要是一些说明信息，类似于定义该 Index 的 schema 信息

  例如指定每个 Field 字段的数据类型

拉勾教育

Index Template（模板）

拉勾教育

- order

  主要作用于在多个 Index Template 同时匹配到一个 Index 的情况

  如果此时这些Index Template 中的配置出现不一致，则以 order 的最大值为准，order 默认值为 0

  创建 Index 的命令中如果自带了 settings 或 mappings 配置，则其优先级最高

- aliases

  是为匹配的 Index 创建别名

  可以通过请求 http://localhost:9200/_alias/* 获取所有别名与 Index 之间的对应关系

拉勾教育

segment 模板

```
{
  "segment":{
    "order":0,
    "index_patterns":[
      "segment-*"
    ],
    "settings":{
      "index":{
        "refresh_interval":"3s",
        "number_of_shards":"2",
        "number_of_replicas":"0"
        // 省略 analysis字段设置
      }
    },
    "mappings":{
      "type":{
        "properties":{
          "segment_id":{
```

拉勾教育

segment 模板

```
            "segment_id":{
              "type":"keyword"
            },
            "trace_id":{
              "type": "keyword"
            },
            "service_id":{
              "type": "integer"
            },
            // 省略其他字段的设置
          }
        }
      },
      "aliases":{ // 为匹配的index创建别名
        "segment":{}
      }
}
```

拉勾教育

节点角色

一个 ElasticSearch 集群是由一个或多个节点组成

这些节点共同存储了集群中的所有数据，并且 ElasticSearch 提供了跨节点的联合索引和搜索功能

集群名称是一个 ElasticSearch 集群的唯一标识

在同一个网络环境中，需要保证集群名称不重复

否则集群中的节点可能会加入到错误的集群中

拉勾教育
L / A / G / O / U

节点角色

ElasticSearch 集群是去中心化的

ElasticSearch 节点的相互发现是基于 Pull-Push 版本的 Gossip 算法实现的

Zen Discovery 是 ElasticSearch 默认的发现实现

提供了广播和单播的能力，帮助一个集群内的节点完成快速的相互发现

拉勾教育
L / A / G / O / U

节点角色

- Master Eligible Node （候选主节点）：

  可以被选举为 Master 的候选节点

- Master Node （主节点）：

  一般采用 quorum 版本的 Bully 算法变体

  主节点是从候选主节点中选举出来的，主要负责管理 ElasticSearch 集群

  通过广播的机制与其他节点维持关系

  负责集群中的 DDL 操作（创建/删除索引），管理其他节点上的分片

拉勾教育
L / A / G / O / U

节点角色

- Data Node（数据节点）：

  存放数据的节点，负责数据的增删改查

- Coordinating Node（协调节点）：

  每个节点都是一个潜在的协调节点

  协调节点最大的作用就是响应客户端的请求，将各个分片里的数据汇总起来一并返回给客户端

- Ingest Node（提取节点）：

  能执行预处理管道，不负责数据也不负责集群相关的事务

拉勾教育
L / A / G / O / U

分片&副本

在 ElasticSearch 中的一个 Index 可以存储海量的 Document，单台机器的磁盘大小是无法存储的

而且在进行数据检索的时候，单台机器也存在性能瓶颈，无法为海量数据提供高效的检索

ElasticSearch 将单个 Index 分割成多个分片

创建 Index 时，可以按照预估值指定任意数量的分片

单个分片都是一个功能齐全且独立的 Index

一个分片可以被分配到集群中的任意节点上

拉勾教育
L / A / G / O / U

分片&副本

通过分片的功能，Index 有了容量水平扩展的能力

运维人员可以通过添加节点的方式扩充整个集群的容量

在处理检索请求时

不同的分片由不同的 ElasticSearch 节点进行检索，可以实现并发操作

拉勾教育
L / A / G / O / U

分片&副本

拉勾教育

当写入一条 Document 的时候，ElasticSearch 会根据指定的 key 决定其所在的分片编号

```
分片编号 = hash(key) % 主分片数量
```

主分片的数量决定了 Document 所在的分片编号，所以在创建 Index 之后，主分片数量不能改变

在进行搜索时，每个分片产生的部分查询结果，也是由 ElasticSearch 集群负责进行聚合的

拉勾教育

分片&副本

单台服务器在实际使用中可能会因为这样或那样的原因发生故障

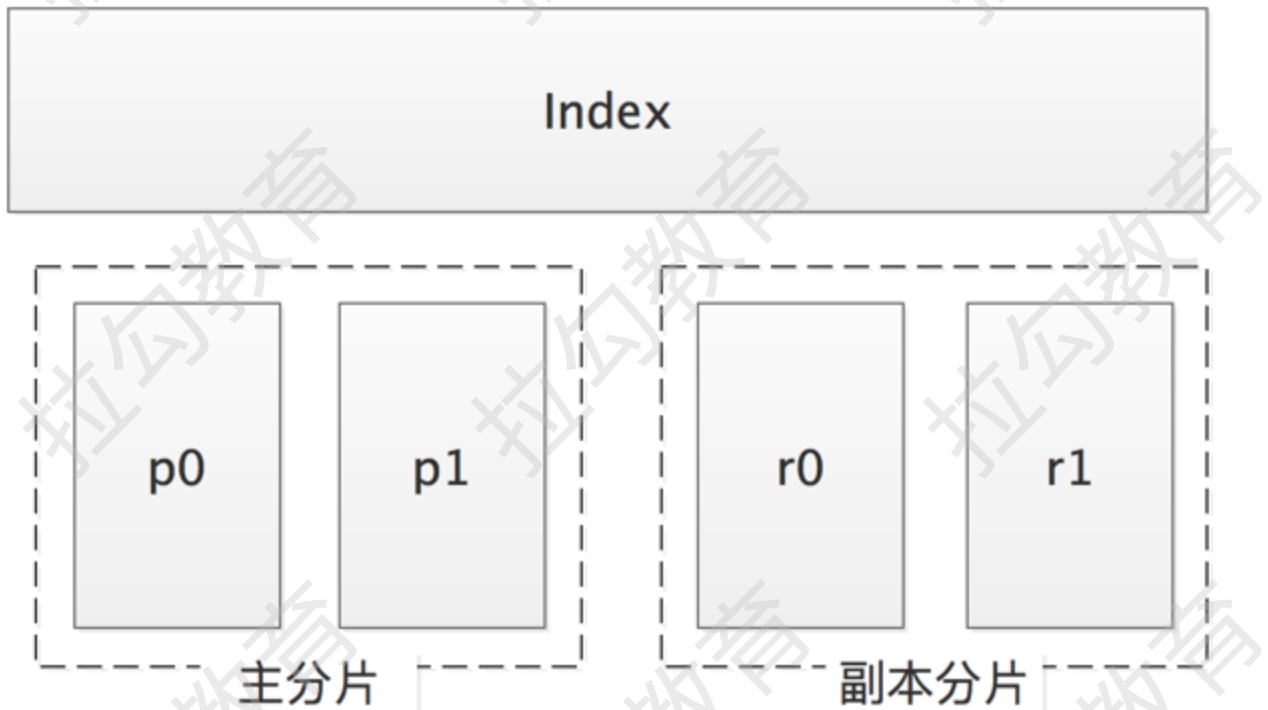
例如意外断电、系统崩溃、磁盘寿命到期等

当发生故障时，该节点负责的分片就无法对外提供服务了

此时需要有一定的**容错机制**，在发生故障时保证此分片可以继续对外提供服务

拉勾教育
L / A / G / O / U

分片&副本



副本带来的好处：

- 在主分片出现故障的时候

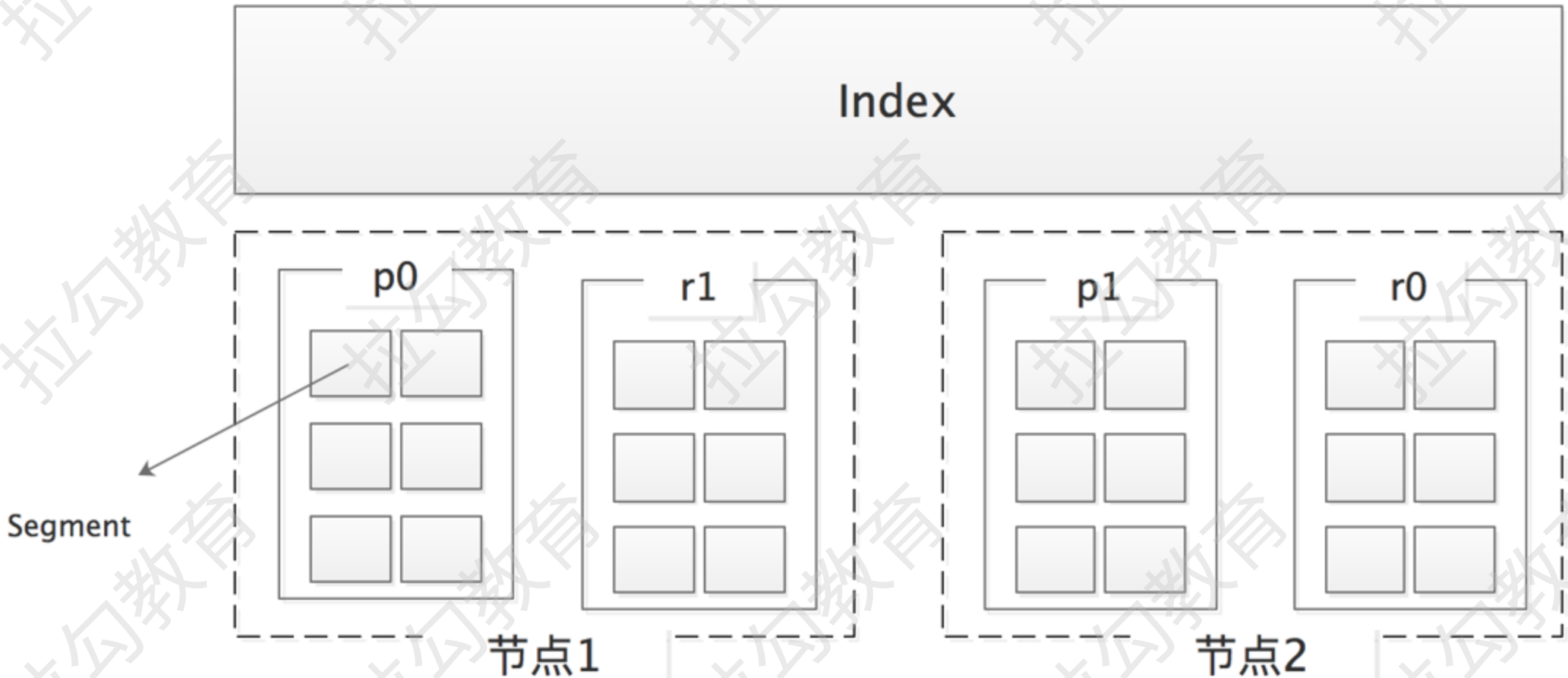
  可以通过副本继续提供服务

- 查询操作可以在分片副本上执行

  可以提升整个 ElasticSearch 查询性能

拉勾教育
L / A / G / O / U

ElasticSearch 写入流程简介



拉勾教育
L / A / G / O / U

ElasticSearch 写入流程简介

拉勾教育

在很多分布式系统中都能看到类似的设计，这种设计的好处：

- 旧 Segment 不支持修改，那么在读操作的时候就不需要加锁，省去了锁本身以及竞争锁相关的开销
- 只有最新的 Segment 支持写入，可以实现顺序写入的效果，增加写入性能
- 只有最新的 Segment 支持写入，可以更好的利用文件系统的 Cache 进行缓存，提高写入和查询性能

拉勾教育

ElasticSearch 集群处理一个写入请求的过程

写入请求会首先发往协调节点（Coordinating Node）

协调节点可能是 Client 连接上的任意一个节点

根据 Document Id 找到对应的主分片所在的节点

拉勾教育
L / A / G / O / U

ElasticSearch 集群处理一个写入请求的过程

拉勾教育

由主分片所在节点处理写入请求

先是写入 Transaction Log

而后将数据写入内存中，默认情况下每隔一秒会同步到 FileSystem Cache 中

默认情况下每隔 30s，会将 FileSystem cache 中的数据写入磁盘中

为了降低数据丢失的概率，可以将这个时间缩短，甚至设置成同步的形式

拉勾教育

ElasticSearch 集群处理一个写入请求的过程

拉勾教育

可以设置三种副本写入策略：

- quorum：默认为 quorum 策略，即超过半数副本写入成功之后，相应写入请求即可返回给客户端
- one ：one 策略是只要成功写入一个副本，即可向客户端返回
- all：all 策略是要成功写入所有副本之后，才能向客户端返回

拉勾教育

ElasticSearch 集群处理一个写入请求的过程

ElasticSearch 的删除操作只是逻辑删除

在每个 Segment 中都会维护一个 .del 文件

删除操作会将相应 Document 在 .del 文件中标记为已删除

查询时依然可以查到

但会在结果中将这些"已删除"的 Document 过滤掉

拉勾教育

L / A / G / O / U

ElasticSearch 集群处理一个写入请求的过程

拉勾教育

由于旧 Segment 文件无法修改

ElasticSearch 会将旧版本的 Document 在 .del 文件中标记为已删除

将新版本的 Document 索引到最新的 Segment 中

拉勾教育

ElasticSearch 集群处理一个写入请求的过程

随着数据的不断写入，将产生很多小 Segment 文件

ElasticSearch 会定期进行 Segment Merge

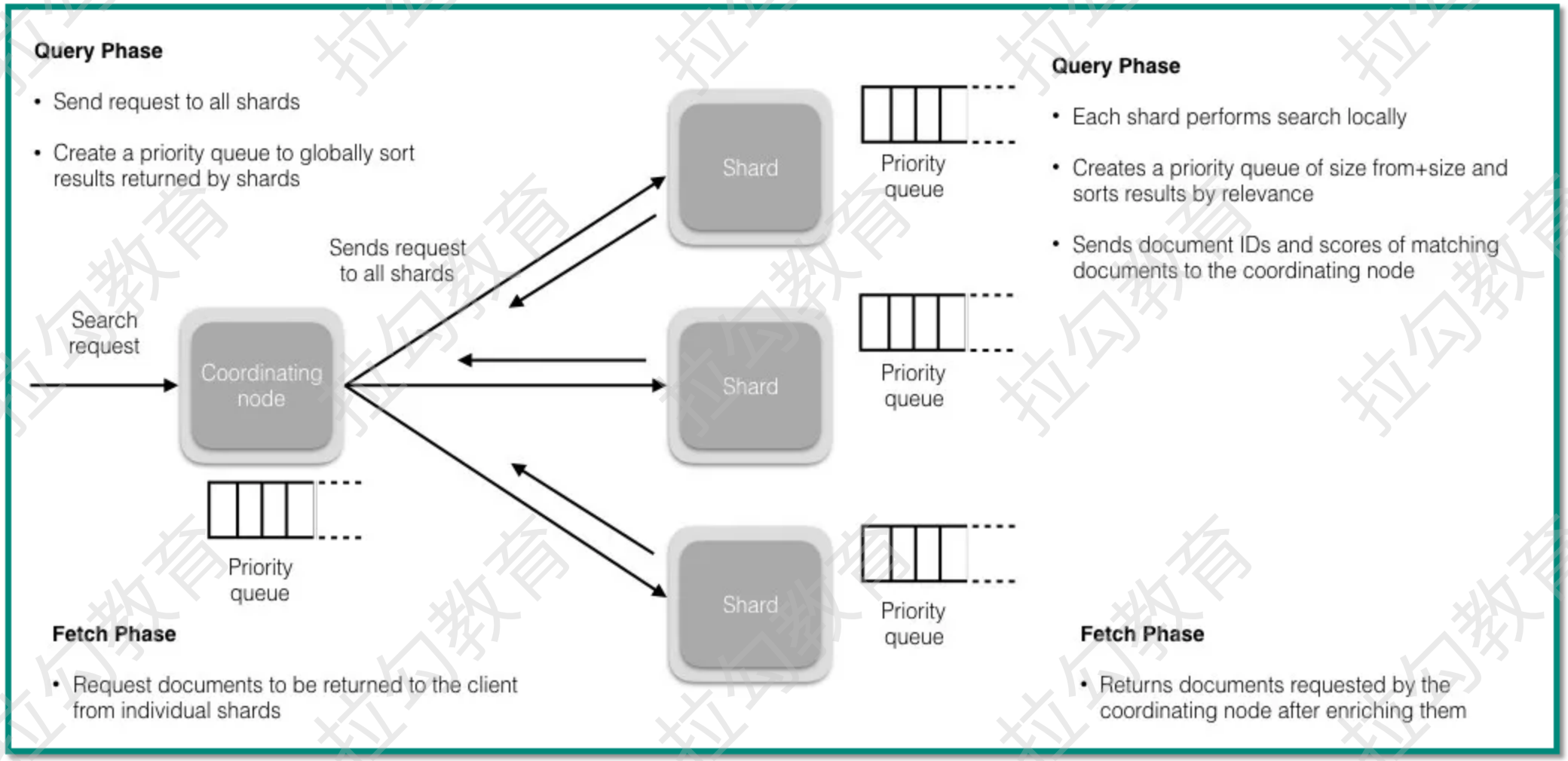
从而减少碎片文件，降低文件打开数，提升 I/O 性能

在 Merge 过程中可以同时根据 .del 文件

将被标记的 Document 真正删除，此时才是真正的物理删除

拉勾教育
L / A / G / O / U

ElasticSearch 查询流程简介



拉勾教育
L / A / G / O / U

High Level REST Client 入门

拉勾教育

ElasticSearch 提供了两种 Java Client，Low Level REST Client 和 High Level REST Client

两者底层都是通过 HTTP 接口与 ElasticSearch 进行交互的：

- Low Level REST Client 需要使用方自己完成请求的序列化以及响应的反序列化；
- High Level REST Client 是基于 Low Level REST Client 实现的

  调用方直接使用特定的请求/响应对象即可完成数据的读写

  High Level REST Client 提供的 API 都会有同步和异步(async 开头)两个版本

  - 同步方法直接返回相应的 response 对象
  - 异步方法需要添加相应的 Listener 来监听并处理返回结果

拉勾教育 L / A / G / O / U

High Level REST Client 入门

SkyWalking 中提供的 ElasticSearchClient 是对 High Level REST Client 的封装

本课时将简单介绍 High Level REST Client 的基本操作

更加完整的 API 使用可以参考 ElasticSearch 官方文档

拉勾教育
L / A / G / O / U

总结

- 介绍 ElasticSearch 中的核心概念，并将这些概念与数据库进行了对比
- 介绍 ElasticSearch 集群中各个节点的角色，以及 ElasticSearch 引入分片和副本功能要解决的问题
- 对 ElasticSearch 读写数据的核心流程进行概述
- 通过示例的方式介绍了 ElasticSearch High Level REST Client 的基本使用方式

拉勾教育
L / A / G / O / U

拉勾教育

Next：第08讲《 Skywalking 源码环境搭建》

L / A / G / O / U

拉勾教育

—互联网人实战大学—

关注拉勾「教育公众号」
获取更多课程信息

拉勾教育