

Elasticsearch的代码结构和写入查询 流程的解读

张磊，极限科技 | 搜索引擎研发负责人

2024-08-28

分享嘉宾



张磊，极限科技 | 搜索引擎研发负责人

对 Elasticsearch 和 Lucene 源码比较熟悉，目前主要负责公司的 Easysearch 产品的研发以及客户服务工作。

Agenda

一、深入解析 Elasticsearch 代码结构

- 项目的整体结构
- 核心模块及其职责
- 插件系统的设计

Agenda

二、详细解读 Elasticsearch 写入和查询流程

- 文档索引过程的各个阶段
- 查询解析和执行的步骤
- 分布式查询的协调过程

项目整体结构

核心组件：

- server: Elasticsearch 服务器的核心实现
- modules: 包含各种功能模块
- plugins: 插件模块

客户端和API：

- client: rest客户端库实现
- rest-api-spec: REST API 规范定义

构建和开发工具：

- buildSrc: 自定义 Gradle 插件和任务，依赖管理
- gradle: Gradle 构建系统配置
- dev-tools: 代码和文档的同步和提交工具

测试和质量保证：

- test: 测试框架
- qa: 质量保证和集成测试
- benchmarks: 性能基准测试

项目整体结构

文档和许可:

- docs: 项目文档
- licenses: 许可证文件

分发和打包:

- distribution: 构建和分发策略, 支持多种操作系统、架构

扩展功能:

- x-pack: 商业扩展功能包

辅助目录:

- .ci: 持续集成配置
- .github: GitHub 相关配置
- .idea: IDE (IntelliJ IDEA) 配置

依赖管理:

- libs: es各个模块的公共依赖项

核心模块职责

数据处理核心:

- Action模块: 处理客户端请求
- Index模块: 管理索引操作
- Search模块: 实现搜索功能

集群管理:

- Cluster模块: 管理集群状态和配置
- Discovery模块: 节点发现和集群形成
- Transport模块: 节点间通信

核心模块职责

索引管理:

- Indices模块: 多索引操作和状态管理
- Gateway模块: 集群状态持久化

系统和扩展:

- Env模块: 运行环境配置
- Plugins模块: 功能扩展支持
- Monitor模块: 性能监控和统计

基础设施:

- Common模块: 公共工具和基础类

插件系统设计

插件系统非常灵活和强大

插件类型多样化:

| | |
|-------------------|--------------------------|
| ActionPlugin: | 用于添加自定义动作 |
| AnalysisPlugin: | 用于添加自定义分析器 |
| ClusterPlugin: | 用于自定义集群行为 |
| DiscoveryPlugin: | 用于自定义节点发现机制 |
| IngestPlugin: | 用于添加自定义数据预处理管道 |
| MapperPlugin: | 用于添加自定义映射器 |
| NetworkPlugin: | 用于扩展网络和传输相关类 |
| RepositoryPlugin: | 用于添加自定义快照存储库 |
| ScriptPlugin: | 可扩展 Elasticsearch 的脚本功能。 |
| SearchPlugin: | 用于扩展搜索行为 |
| ReloadablePlugin: | 用于创建可在运行时重新加载的插件 |

插件系统设计

基于接口的扩展点： 每种插件类型都有对应的接口，开发者通过实现这些接口来提供自定义功能

生命周期管理： 可实现 LifecycleComponent 接口，ES 会自动管理这些组件的生命周期

资源隔离： 每个插件都在自己的类加载器中运行,防止冲突

标准化的构建和打包：

ES 提供了标准的方法来构建和打包插件，使得插件的分发和安装变得简单。

插件系统设计

自动设置位于 plugins 目录下 Elasticsearch 插件的构建 简化开发过程

```
24 // only configure immediate children of plugins dir
25 configure(subprojects.findAll { Project it -> it.parent.path == project.path }) {
26     group = 'org.elasticsearch.plugin'
27     apply plugin: 'elasticsearch.esplugin'
28
29     esplugin {
30         // for local ES plugins, the name of the plugin is the same as the directory
31         name project.name
32
33         licenseFile rootProject.file('licenses/APACHE-LICENSE-2.0.txt')
34         noticeFile rootProject.file('NOTICE.txt')
35     }
36 }
37 |
```

完整流程

[完整bulk流程图地址](#)

[完整search流程图地址](#)

写入流程

Elasticsearch Bulk 写入流程

客户端发起 Bulk 请求： 客户端将多个写操作（例如索引、更新、删除）打包成一个 Bulk 请求发送到 Elasticsearch 集群。

协调节点分发请求： 协调节点将 Bulk 请求分解成多个分片请求，并将它们转发到对应的数据节点。

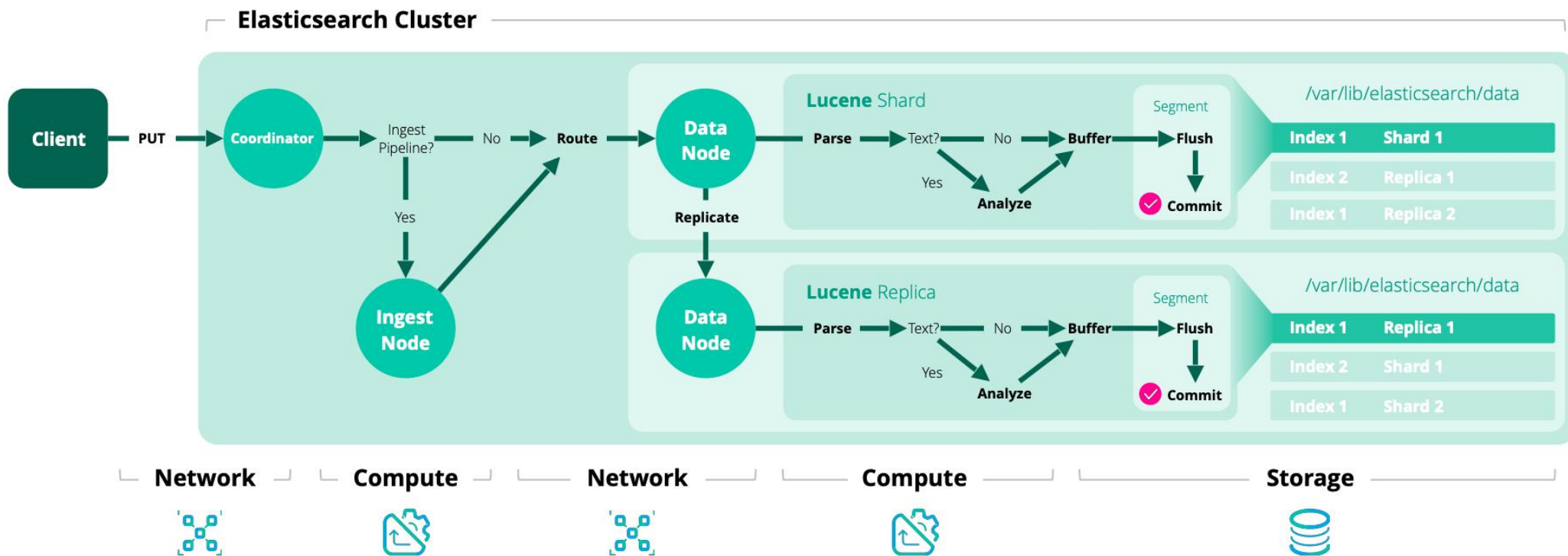
主分片写入数据： 主分片执行写操作，并将数据写入到事务日志和索引中。

副本分片同步数据： 主分片将数据复制到副本分片，确保数据冗余和高可用性，如果有多个副本，并行完成。

响应客户端： 一旦所有同步副本都成功执行了操作并响应了主分片，主分片就向客户端确认请求成功完成。

写入流程

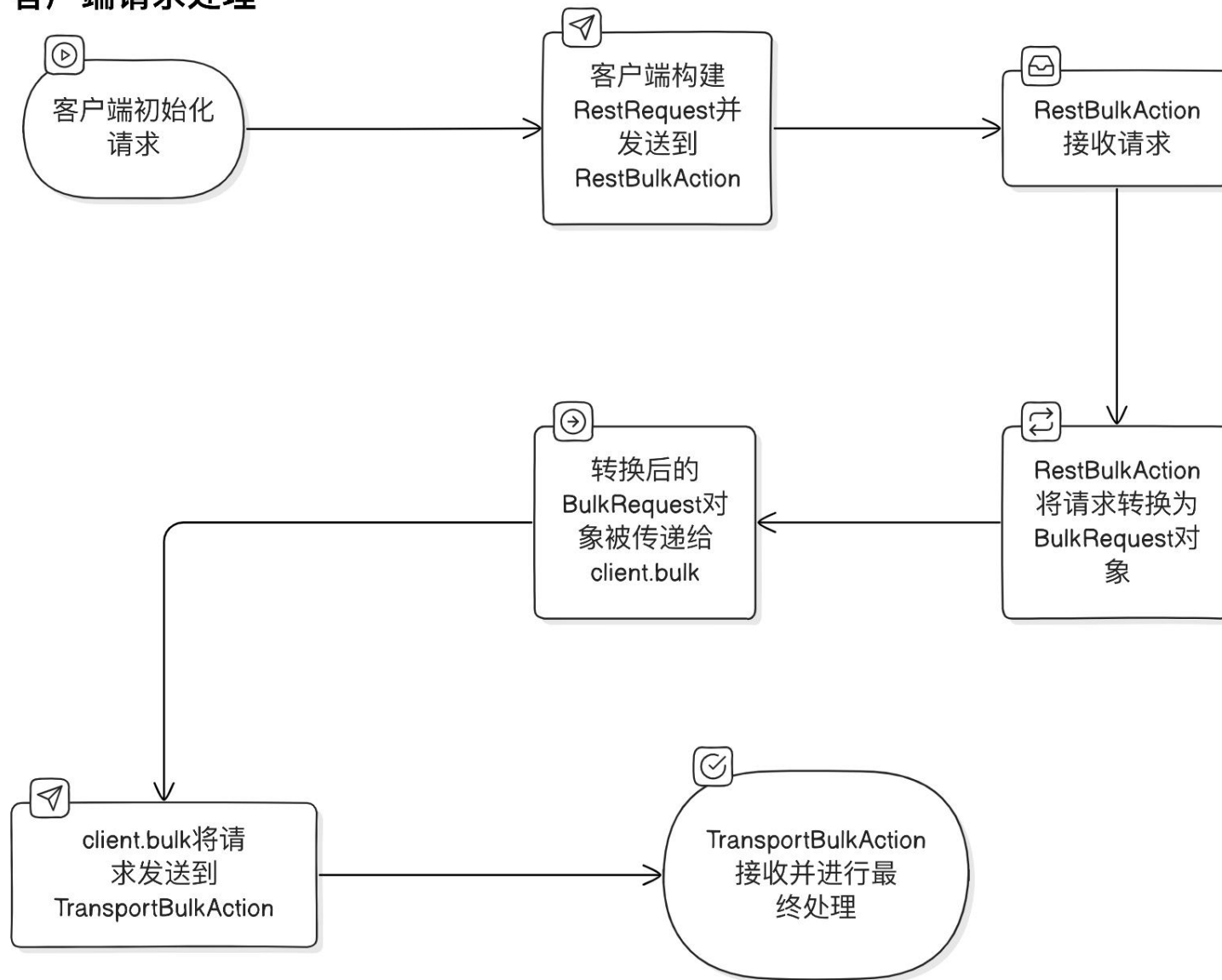
Elasticsearch Bulk 写入流程



详细流程

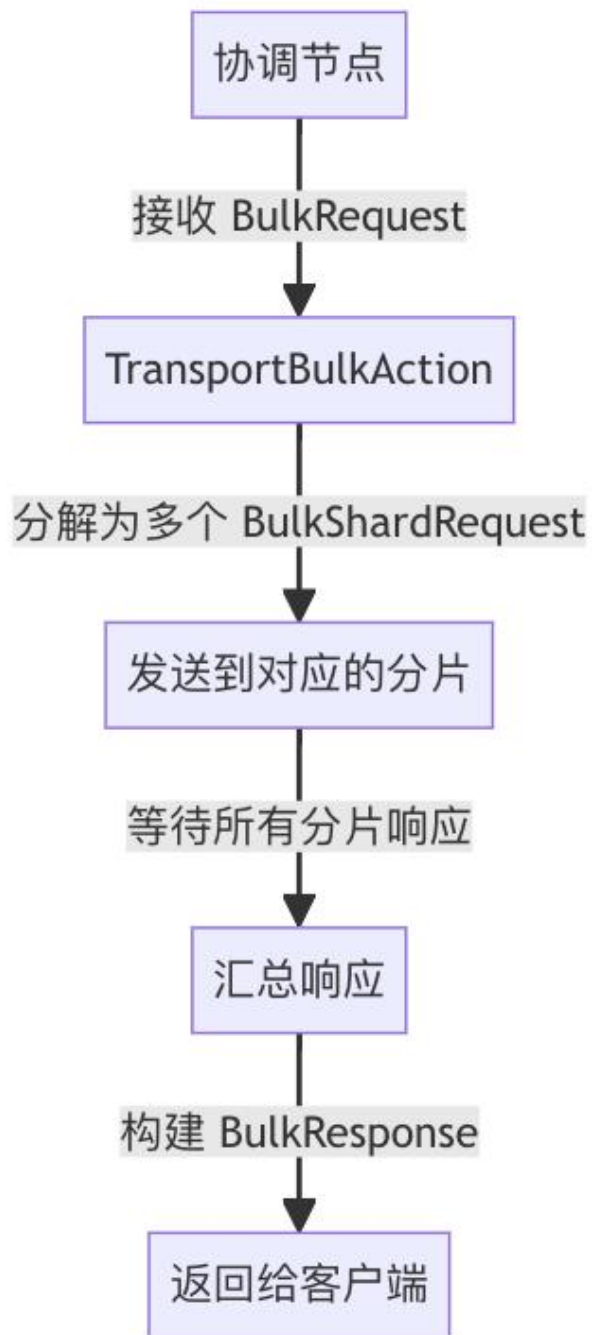
客户端请求处理

客户端请求处理



详细流程

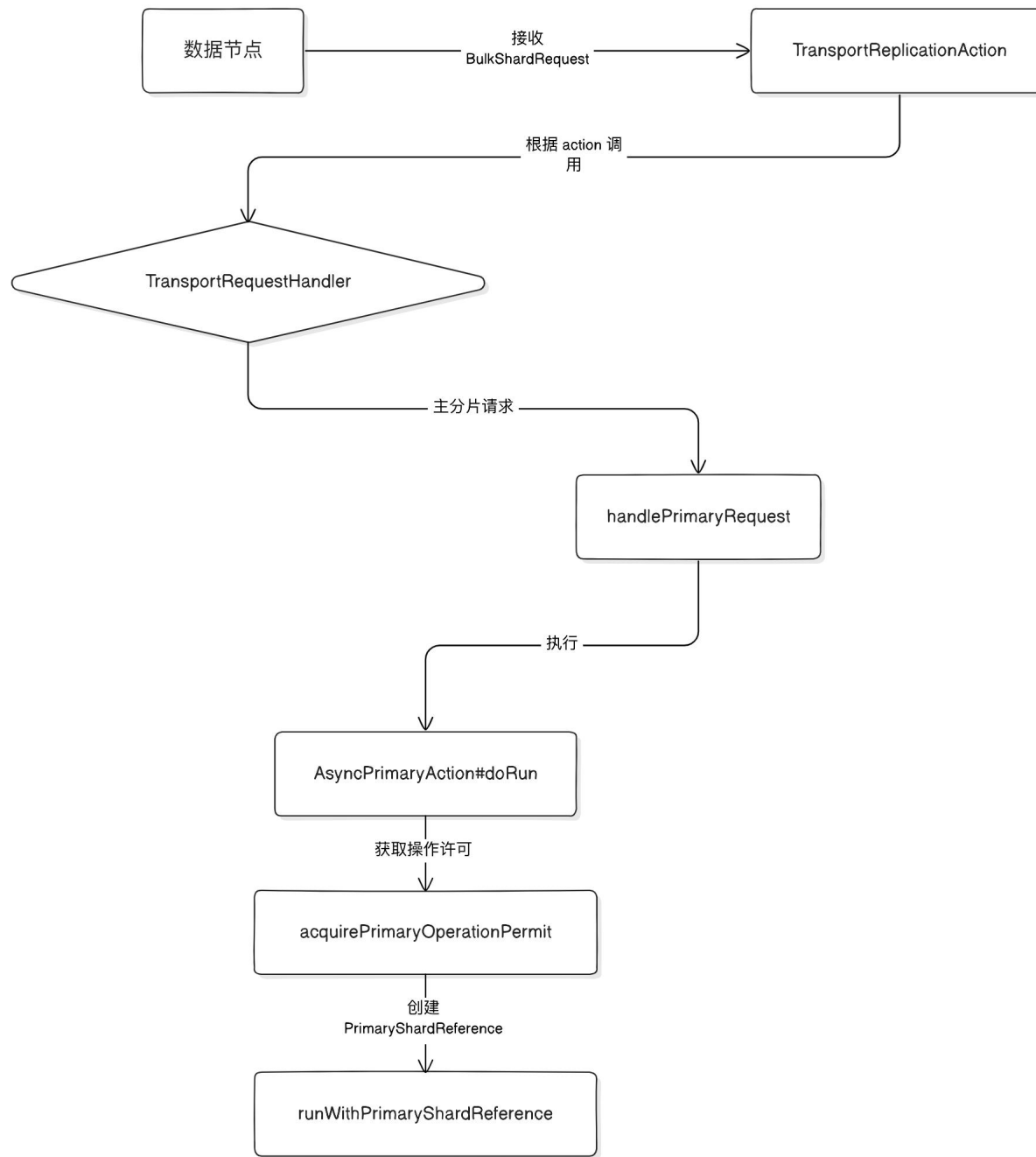
协调节点处理



详细流程

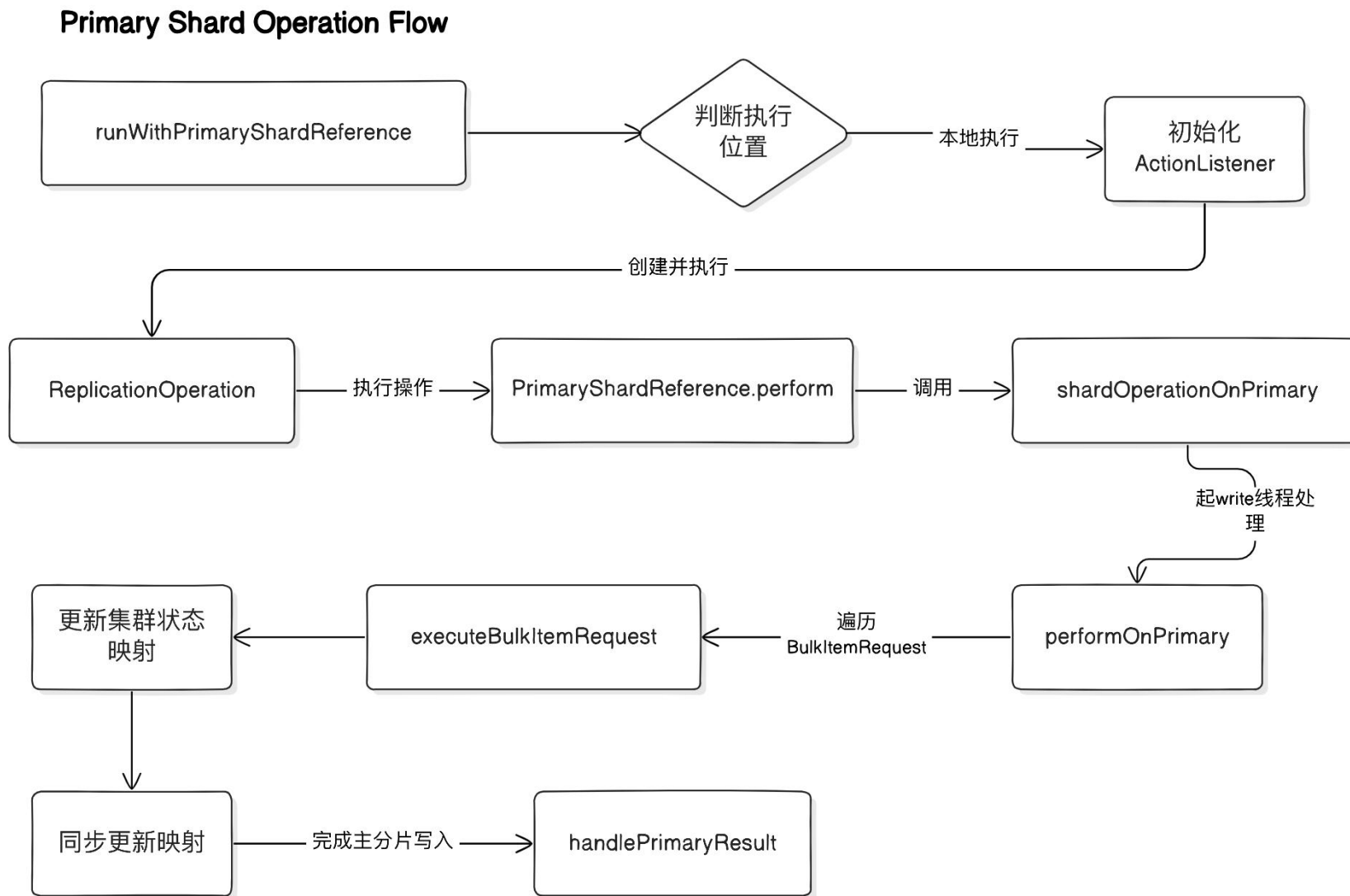
接收主分片写入请求并处理

数据节点处理流程



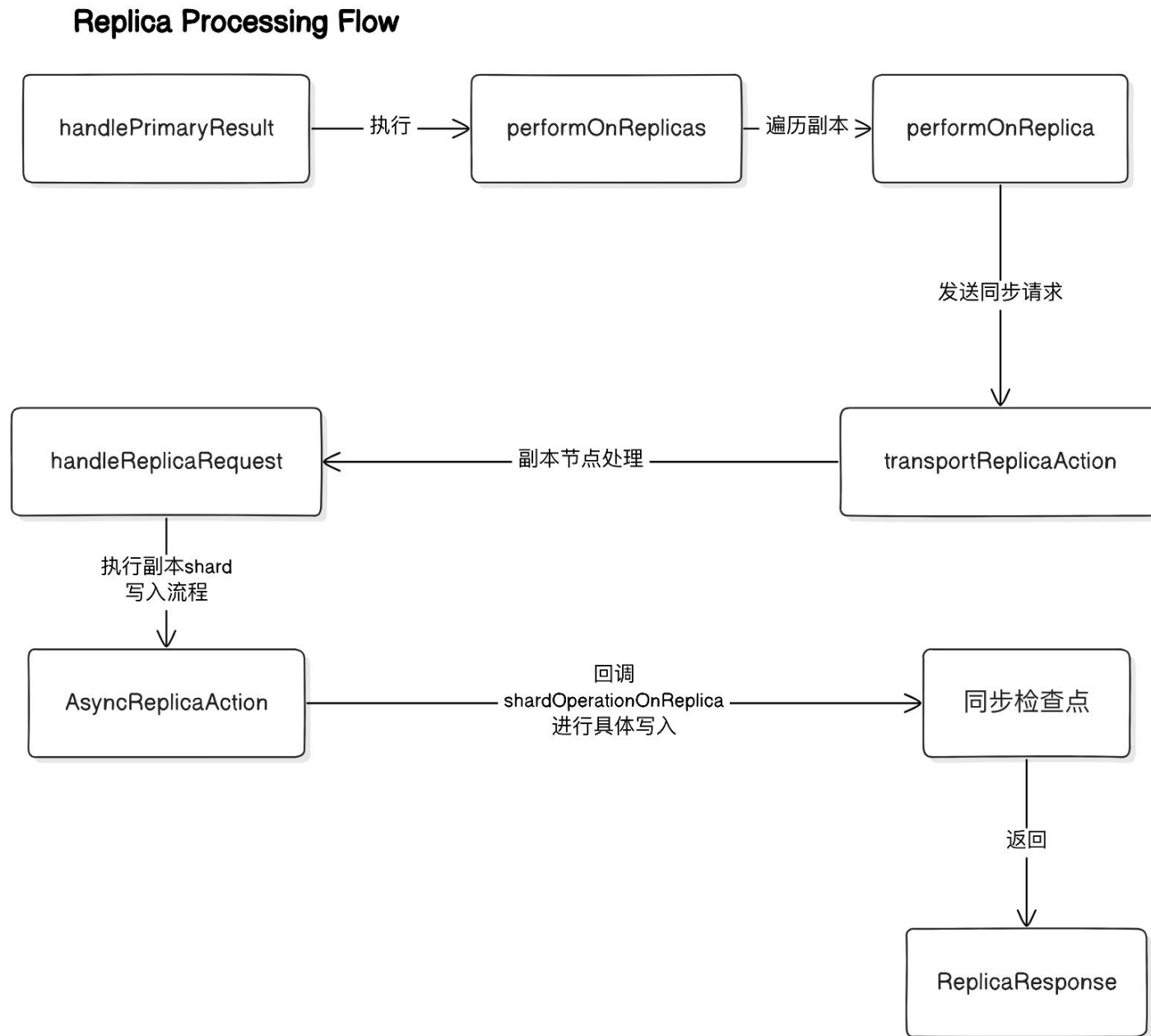
详细流程

主分片节点写入操作



详细流程

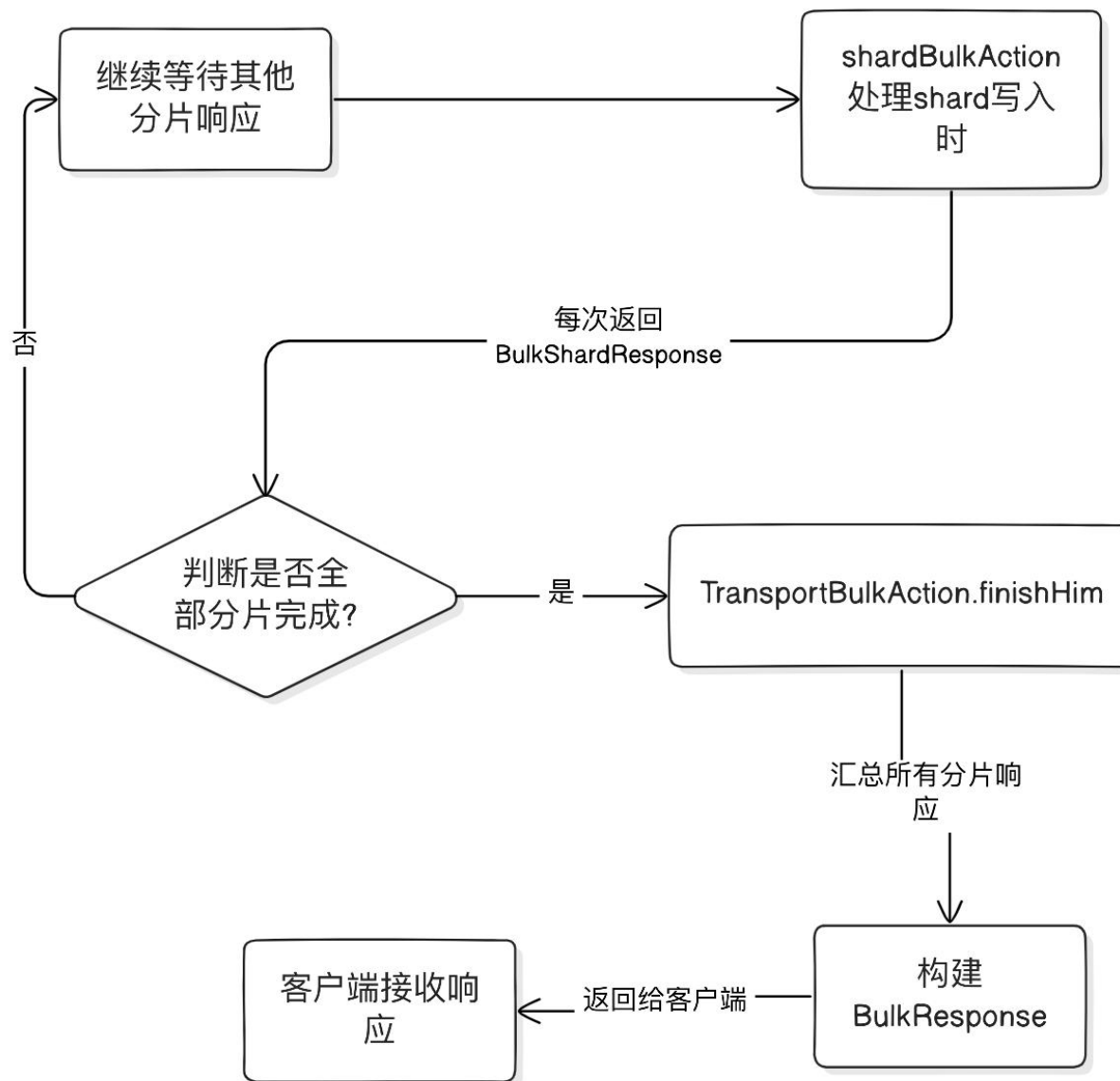
副本同步流程



详细流程

响应处理

Response Processing Flow



搜索流程

客户端发送搜索请求： 客户端将搜索请求发送到 Elasticsearch 集群中的任意一个节点，该节点将充当协调节点。

协调节点解析请求： 协调节点解析搜索请求，确定要搜索的索引、查询条件、排序规则、分页信息等。

分片路由： 协调节点根据路由规则，确定哪些分片包含了需要搜索的数据，并将查询请求转发到这些分片的主分片或副本分片。

分片查询： 每个分片节点执行查询操作，并返回 Top N 个评分最高的文档及其相关性评分。

结果合并： 协调节点将来自各个分片的 Top N 结果合并成全局的 Top N 结果，并根据排序规则进行排序。

搜索流程

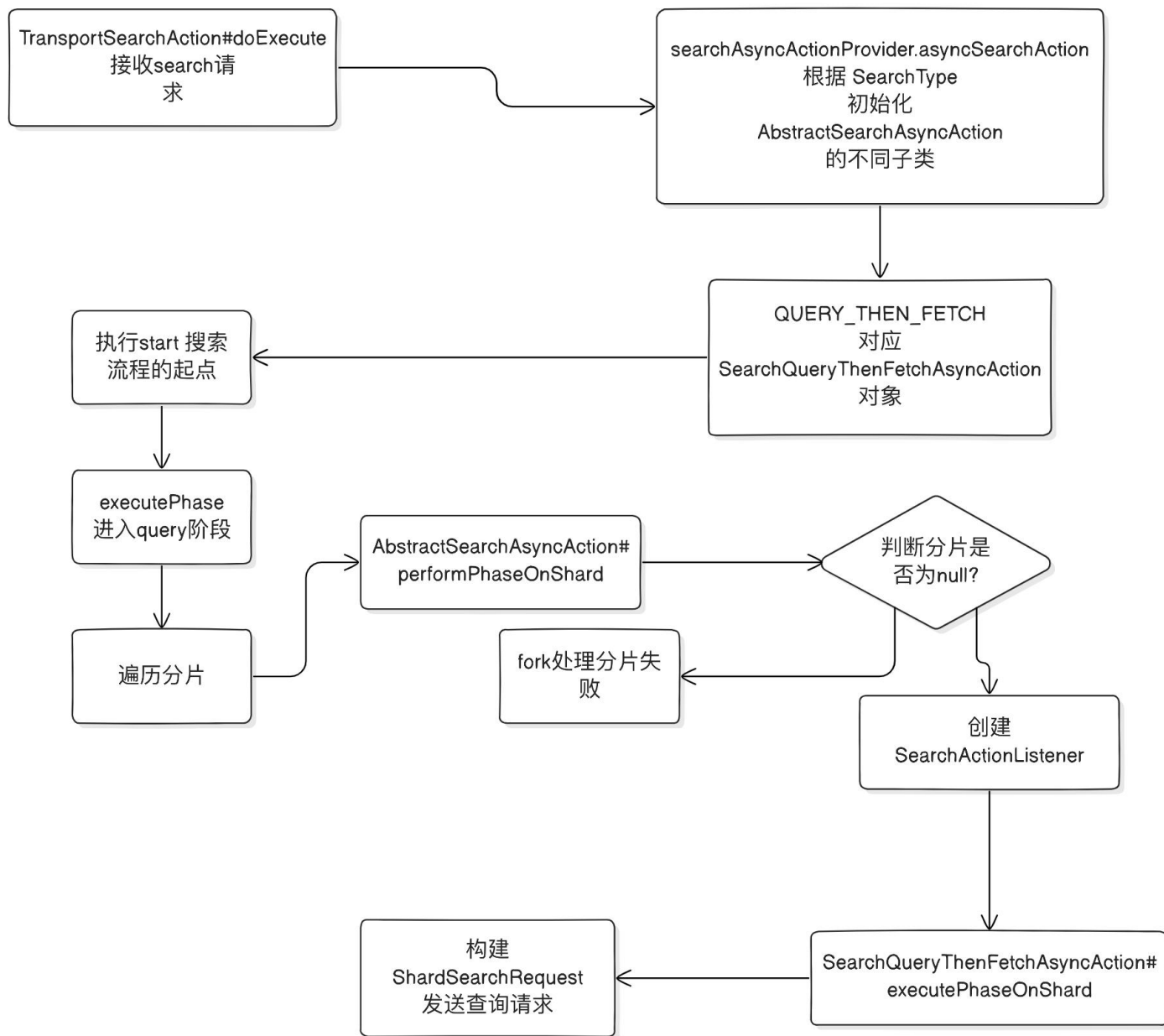
获取文档： 协调节点根据合并后的 Top N 结果，向相关分片节点发送获取请求，获取完整的文档数据。

结果组装： 协调节点将来自不同分片的文档数据组装成最终的搜索结果，包括文档的元数据、高亮片段等。

返回结果： 协调节点将搜索结果返回给客户端。

详细流程

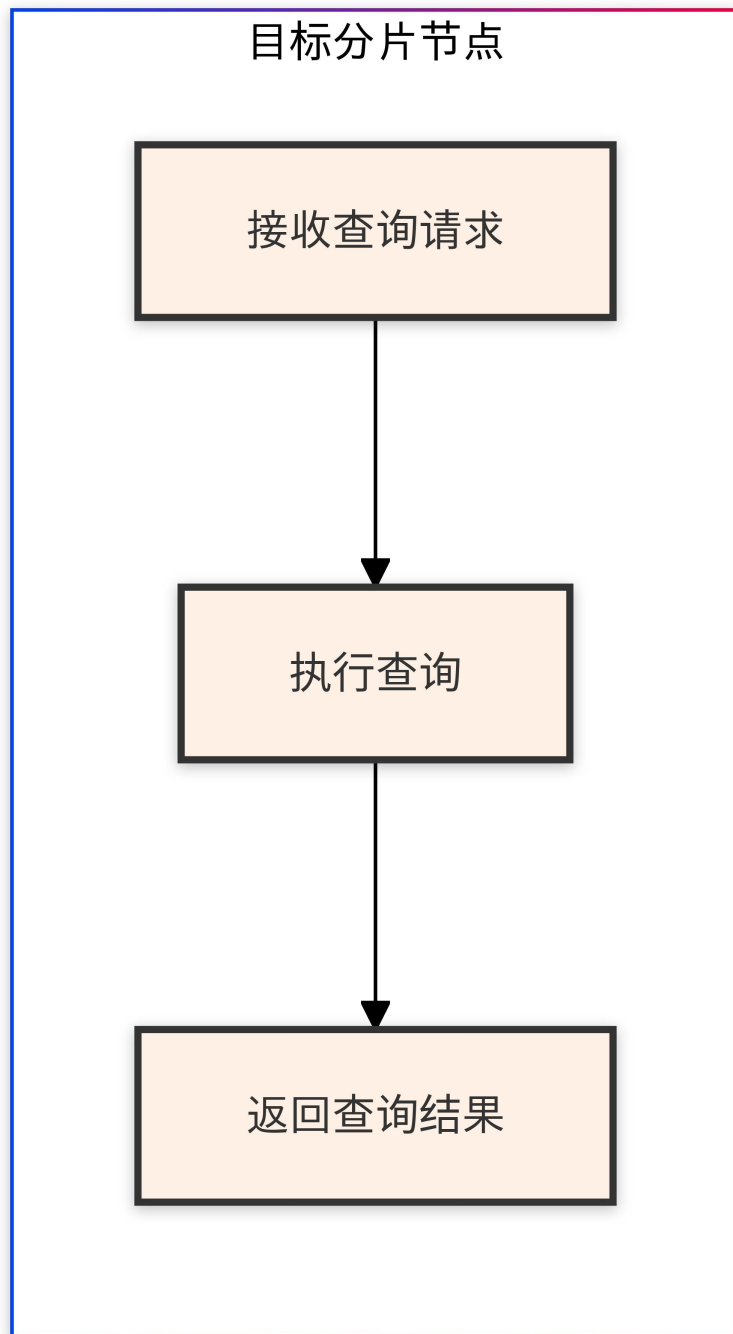
协调节点接收查询请求



协调节点接收查询请求

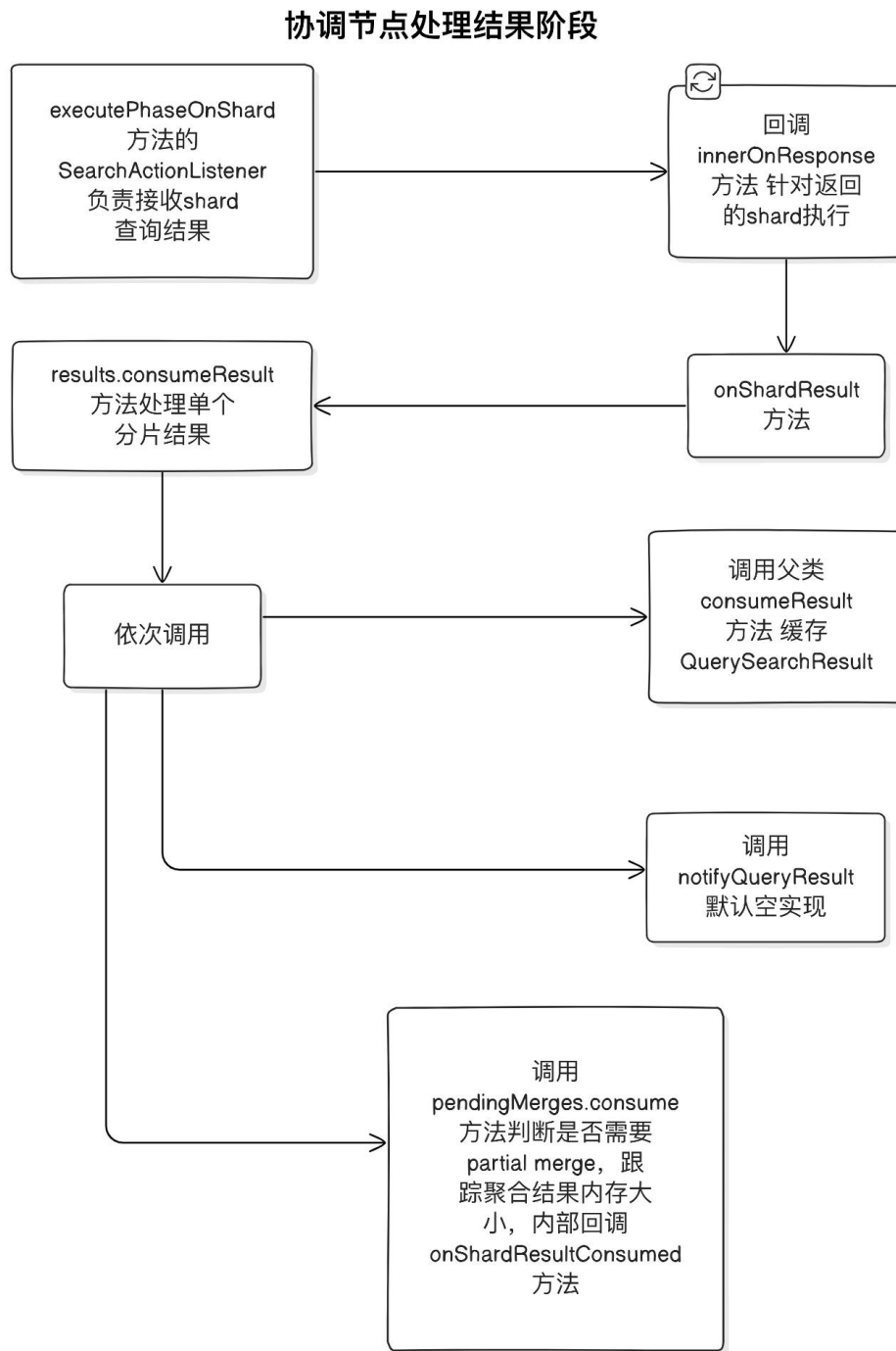
详细流程

目标分片节点处理



详细流程

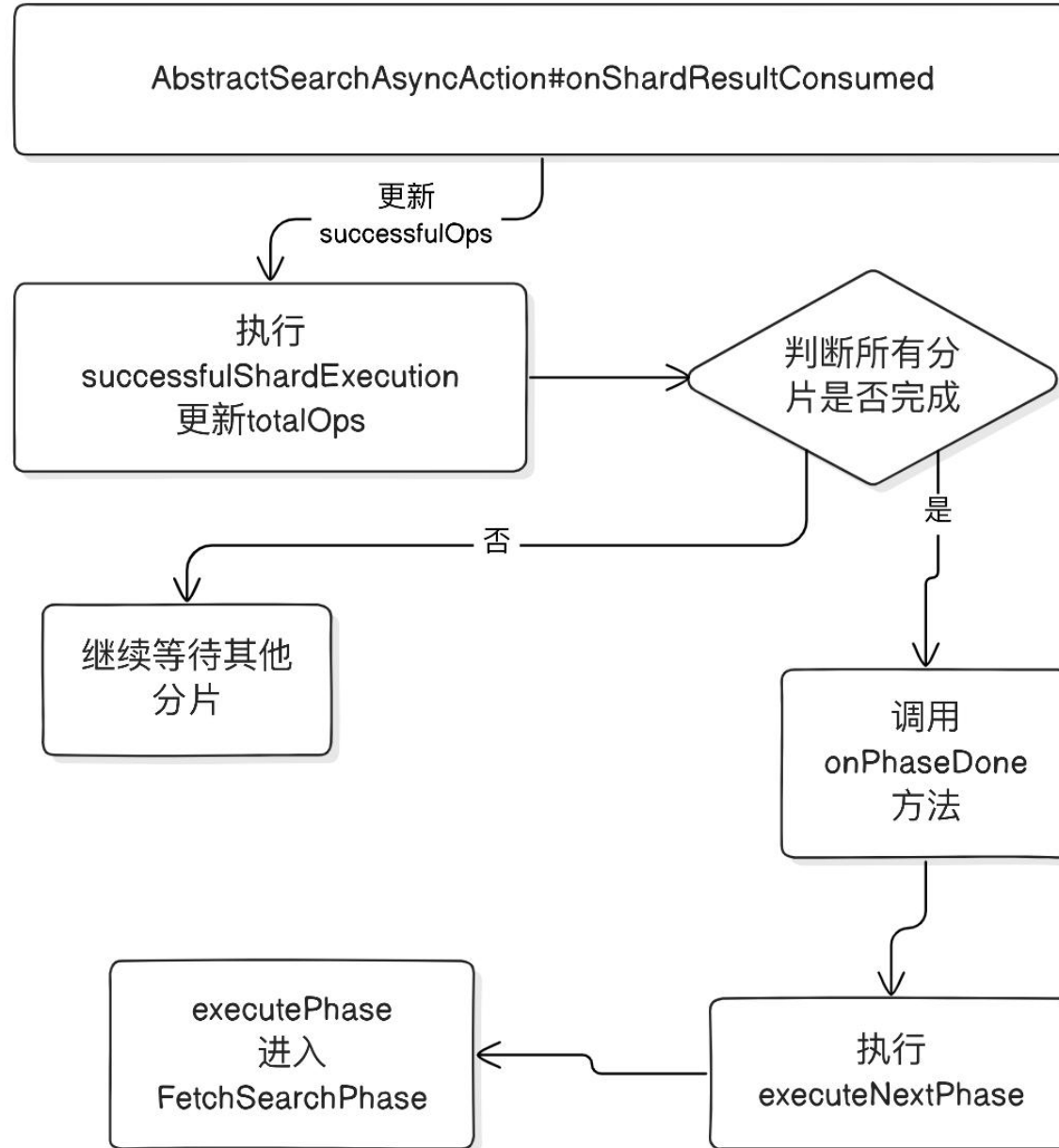
协调节点处理单个分片查询结果



结果处理和阶段转换阶段

详细流程

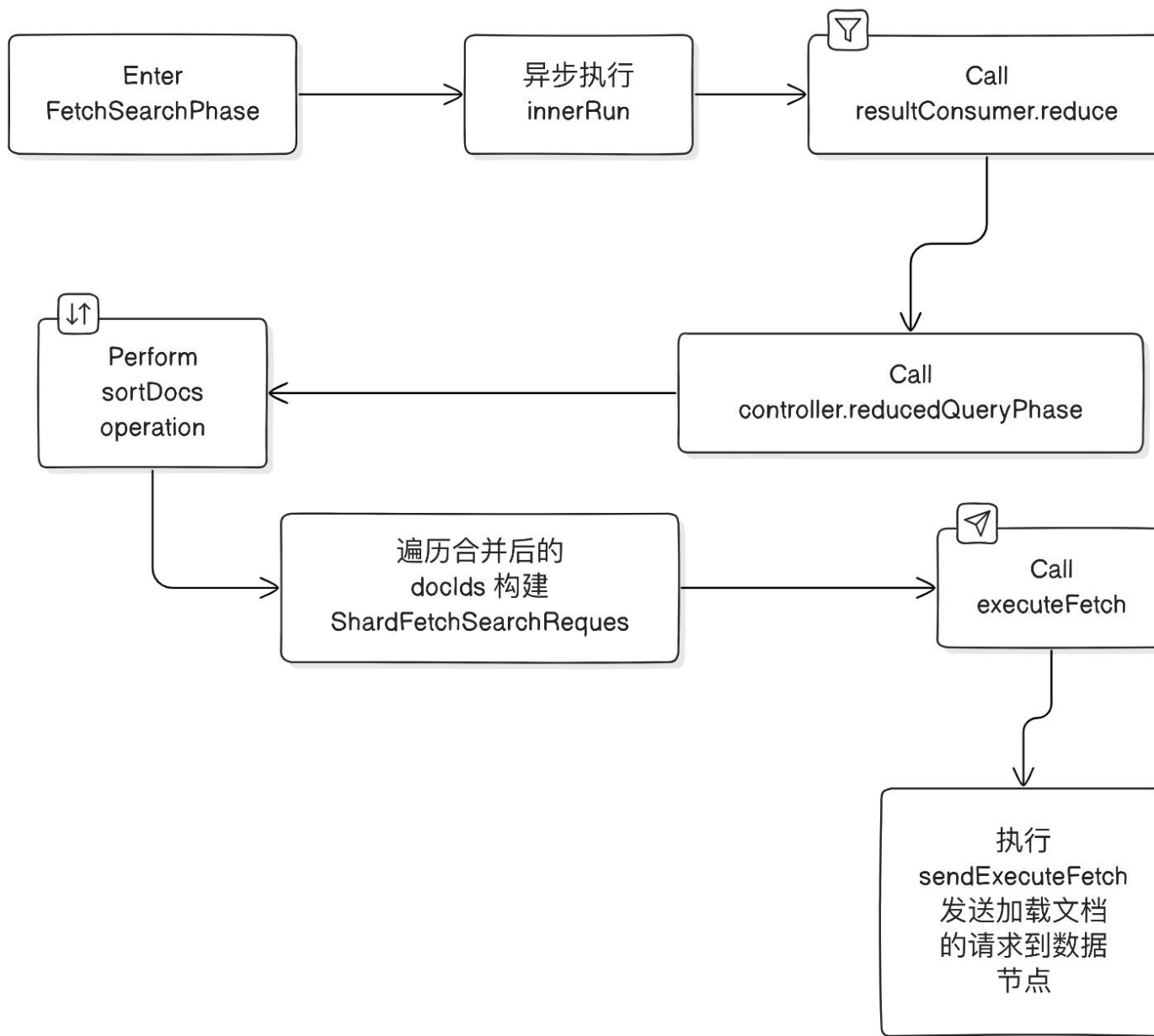
协调节点处理单个分片查询结果



详细流程

Fetch阶段合并
和加载文档

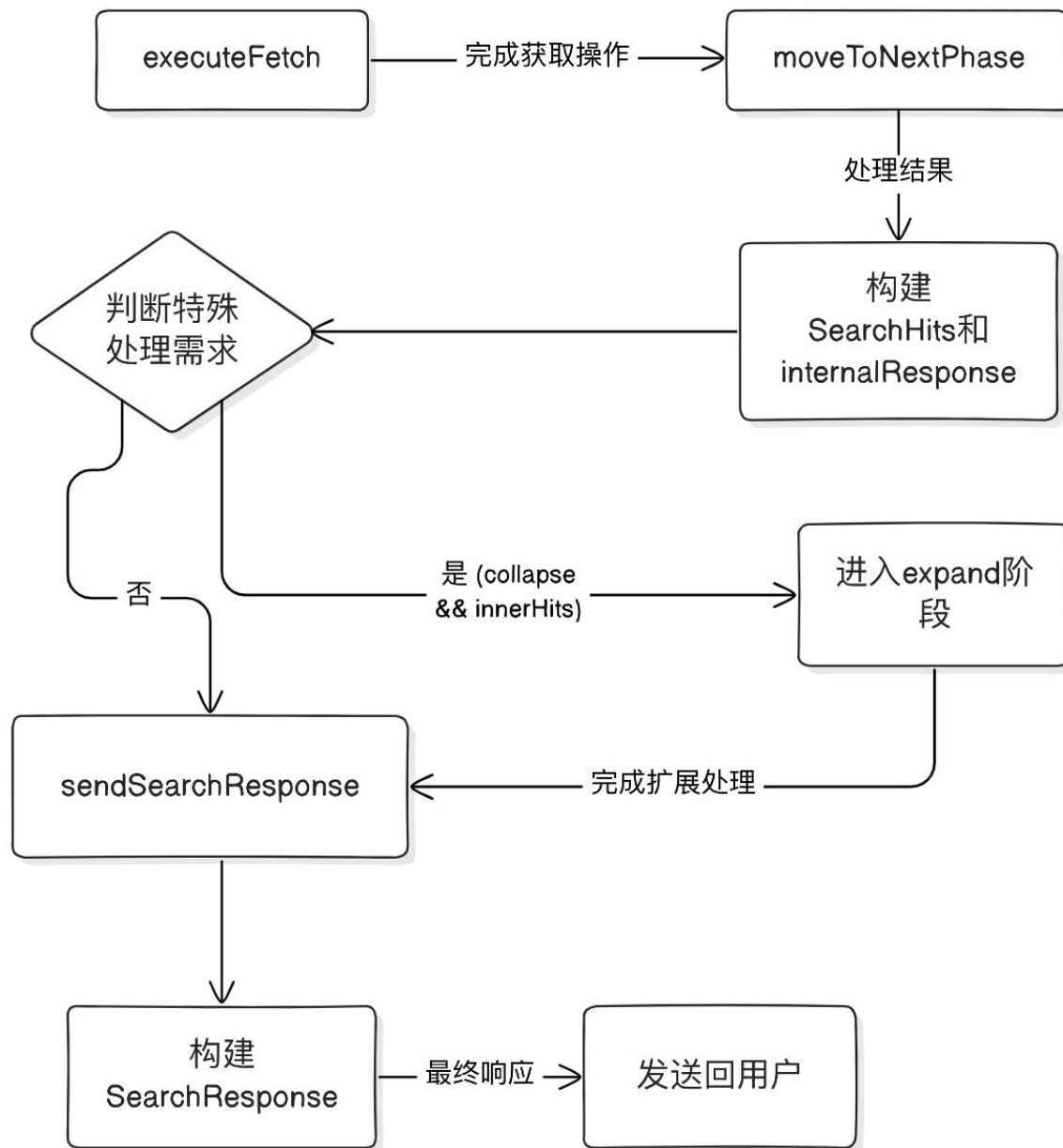
Fetch Phase Execution Flow



最终处理和响应

详细流程

expand 阶段
构建 SearchResponse
返回给客户端





极限科技 · 让搜索更简单

THANK YOU FOR WATCHING

感谢聆听



搜索客 SearchKit

搜索人自己的社区

专业、垂直、纯粹的开源搜索技术交流社区

<https://searchkit.cn>

