



黑马程序员™
www.itheima.com

传智播客旗下
高端IT教育品牌

ElasticSearch 高级

目录 Contents

- ◆ ElasticSearch 高级操作
- ◆ ElasticSearch 集群管理

ElasticSearch 高级操作

- 批量操作
- 导入数据
- 各种查询
- 索引别名和重建索引

批量操作-脚本

Bulk 批量操作是将文档的增删改查一些列操作，通过一次请求全都做完。减少网络传输次数。

语法：

```
POST /_bulk
{"action": {"metadata"}}
{"data"}
```

示例：

```
POST _bulk
{"delete":{"_index":"person", "_id":"5" }}
{"create":{"_index":"person", "_id":"5" }}
{"name":"六号","age":20,"address":"北京"}
{"update":{"_index":"person", "_id":"2" }}
{"doc":{"name":"二号"}}
```

批量操作-JavaAPI

导入数据



案例：需求

将数据库中Goods表的数据导入到ElasticSearch中



案例：实现步骤

- ① 创建goods索引
- ② 查询Goods表数据
- ③ 批量添加到ElasticSearch中

ElasticSearch 高级操作

- 批量操作
- 导入数据
- 各种查询
- 索引别名和重建索引

matchAll查询-脚本

matchAll查询：查询所有文档

语法

```
GET 索引名称/_search
{
  "query": {
    | "match_all": {}
  }
}
```


matchAll查询-JavaAPI

term查询-脚本

term查询：不会对查询条件进行分词。

语法

```
GET 索引名称/_search
{
  "query": {
    "term": {
      "字段名称": {
        "value": "查询条件"
      }
    }
  }
}
```

term查询-JavaAPI

term查询：不会对查询条件进行分词。

match查询-脚本

match查询:

- 会对查询条件进行分词。
- 然后将分词后的查询条件和词条进行等值匹配
- 默认取并集 (OR)

语法

```
GET 索引名称/_search
{
  "query": {
    "match": {
      "字段名称": "查询条件"
    }
  }
}
```

```
GET 索引名称/_search
{
  "query": {
    "match": {
      "字段名称": {
        "query": "查询条件",
        "operator": "操作 (or或and)"
      }
    }
  }
}
```

match查询-JavaAPI

模糊查询-脚本

wildcard查询：会对查询条件进行分词。还可以使用通配符？（任意单个字符）和 *（0个或多个字符）

regexp查询：正则查询

prefix查询：前缀查询

模糊查询-JavaAPI

范围查询-脚本

range 范围查询：查找指定字段在指定范围内包含值

范围查询-JavaAPI

queryString查询-脚本

queryString:

- 会对查询条件进行分词。
- 然后将分词后的查询条件和词条进行等值匹配
- 默认取并集 (OR)
- 可以指定多个查询字段

```
GET 索引名称/_search
{
  "query": {
    "query_string": {
      "fields": ["字段1", "字段2"...],
      "query": "查询条件1 OR 查询条件2"
    }
  }
}
```

queryString查询-JavaAPI

布尔查询-脚本

boolQuery: 对多个查询条件连接。连接方式:

- must (and) : 条件必须成立
- must_not (not) : 条件必须不成立
- should (or) : 条件可以成立
- filter: 条件必须成立, 性能比must高。不会计算得分

```
GET 索引名称/_search
{
  "query": {
    "bool": {
      "must": [ {} ],
      "filter": [ {} ],
      "must_not": [ {} ],
      "should": [ {} ]
    }
  }
}
```

布尔查询-JavaAPI

聚合查询-脚本

- 指标聚合：相当于MySQL的聚合函数。max、min、avg、sum等
- 桶聚合：相当于MySQL的 group by 操作。不要对text类型的数据进行分组，会失败。

聚合查询-JavaAPI

高亮查询-脚本

高亮三要素：

- 高亮字段
- 前缀
- 后缀

```
GET goods/_search
{
  "query": {
    "match": {
      "title": "电视"
    }
  },
  "highlight": {
    "fields": {
      "title": {
        "pre_tags": "<aa>",
        "post_tags": "</aa>"
      }
    }
  }
}
```


高亮查询-JavaAPI

ElasticSearch 高级操作

- 批量操作
- 导入数据
- 各种查询
- 索引别名和重建索引

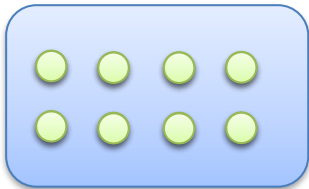
重建索引

随着业务需求的变更，索引的结构可能发生改变。

ElasticSearch的索引一旦创建，只允许添加字段，不允许改变字段。因为改变字段，需要重建倒排索引，影响内部缓存结构，性能太低。

那么此时，就需要重建一个新的索引，并将原有索引的数据导入到新索引中。

student_index_v1



student_index_v2



索引别名

重建索引后，代码中还是使用的老索引在操作ElasticSearch，

1. 改代码（不推荐）
2. 使用别名（推荐）



黑马程序员
www.itheima.com

传智播客旗下高端IT教育品牌

目录 Contents

- ◆ ElasticSearch 高级操作
- ◆ ElasticSearch 集群管理

ElasticSearch 集群管 理

- 集群介绍
 - 搭建集群
 - Kibana 管理集群
 - JavaAPI 访问集群
- 集群原理
- 脑裂
- 集群扩容

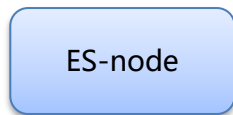
集群介绍



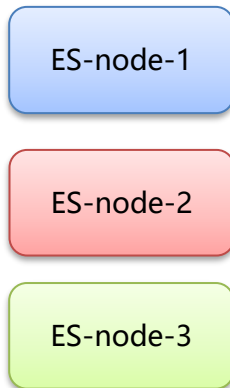
集群和分布式

- 集群：多个人做一样的事。
- 分布式：多个人做不一样的事。

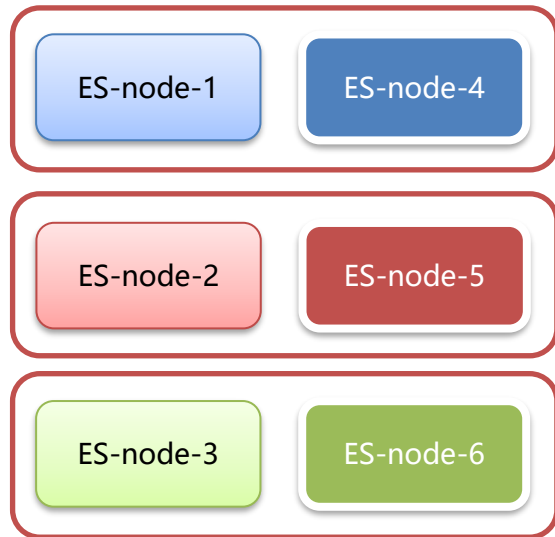
单点架构



集群架构



集群分布式架构



集群解决的问题：

- 让系统高可用
- 分担请求压力

分布式解决的问题：

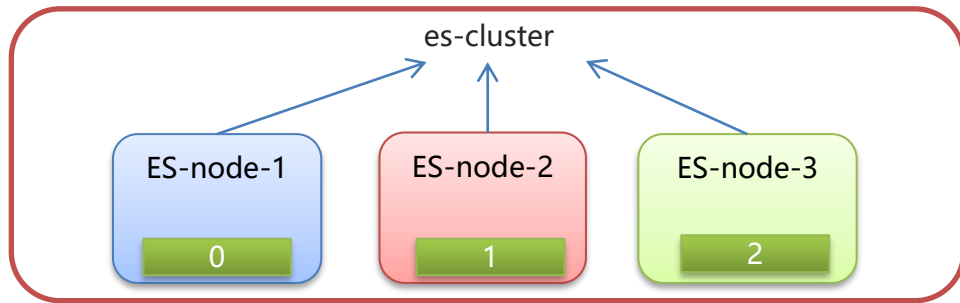
- 分担存储和计算的压力，提速
- 解耦

ElasticSearch 集群特点

- ElasticSearch 天然支持分布式
- ElasticSearch 的设计隐藏了分布式本身的复杂性

ElasticSearch 集群分布式架构相关概念

- 集群 (cluster) : 一组拥有共同的 cluster name 的 节点。
- 节点 (node) : 集群中的一个 Elasticsearch 实例
- 索引 (index) : es存储数据的地方。相当于关系数据库中的database概念
- 分片 (shard) : 索引可以被拆分为不同的部分进行存储, 称为分片。在集群环境下, 一个索引的不同分片可以拆分到不同的节点中
- 主分片 (Primary shard) : 相对于副本分片的定义。
- 副本分片 (Replica shard) 每个主分片可以有一个或者多个副本, 数据和主分片一样。



ElasticSearch 集群管 理

- 集群介绍
- 搭建集群
- Kibana 管理集群
- JavaAPI 访问集群
- 集群原理
- 脑裂
- 集群扩容

ElasticSearch 集群搭建

ElasticSearch 集群管 理

- 集群介绍
- 搭建集群
- Kibana 管理集群
- JavaAPI 访问集群
- 集群原理
- 脑裂
- 集群扩容

JavaAPI 访问集群

```
@Bean
public RestHighLevelClient client(){
    return new RestHighLevelClient(RestClient.builder(
        new HttpHost(
            host,
            port,
            scheme: "http"
        )
    ));
}
```

ElasticSearch 集群管 理

- 集群介绍
- 搭建集群
- Kibana 管理集群
- JavaAPI 访问集群
- 集群原理
- 脑裂
- 集群扩容

集群原理-分片配置

- 在创建索引时，如果不指定分片配置，则默认主分片1，副本分片1。
- 在创建索引时，可以通过settings设置分片

```
"settings": {  
  "number_of_shards": 3,  
  "number_of_replicas": 1  
},
```

- 分片与自平衡：当节点挂掉后，挂掉的节点分片会自平衡到其他节点中
- 在Elasticsearch 中，每个查询在每个分片的单个线程中执行，但是，可以并行处理多个分片。
- 分片数量一旦确定好，不能修改。
- 索引分片推荐配置方案：
 - 每个分片推荐大小10-30GB
 - 分片数量推荐 = 节点数量 * 1~3倍

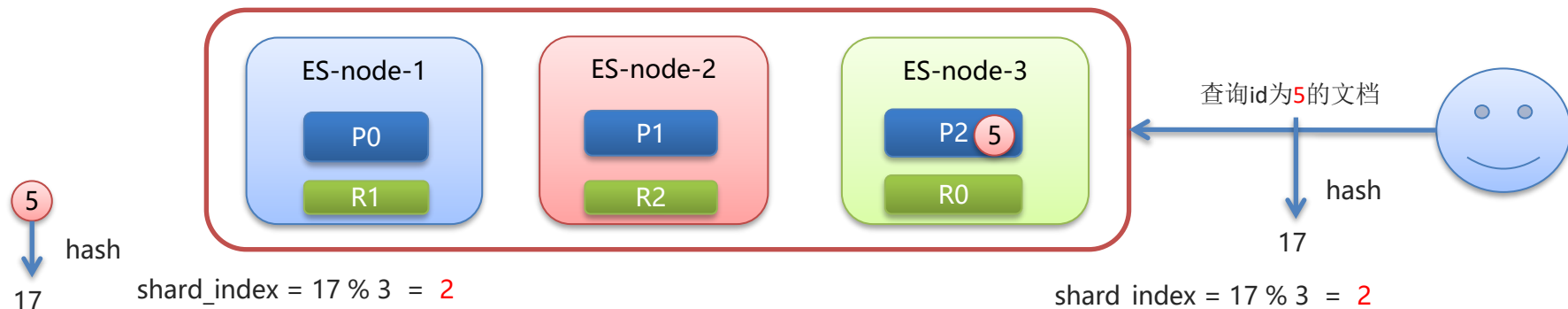
思考：比如有1000GB数据，应该有多少个分片？多少个节点

40个分片

20个节点

集群原理-路由原理

- 文档存入对应的分片，ES计算分片编号的过程，称为路由。
- Elasticsearch 是怎么知道一个文档应该存放到哪个分片中呢？
- 查询时，根据文档id查询文档，Elasticsearch 又该去哪个分片中查询数据呢？
- 路由算法： $\text{shard_index} = \text{hash}(\text{id}) \% \text{number_of_primary_shards}$



ElasticSearch 集群管 理

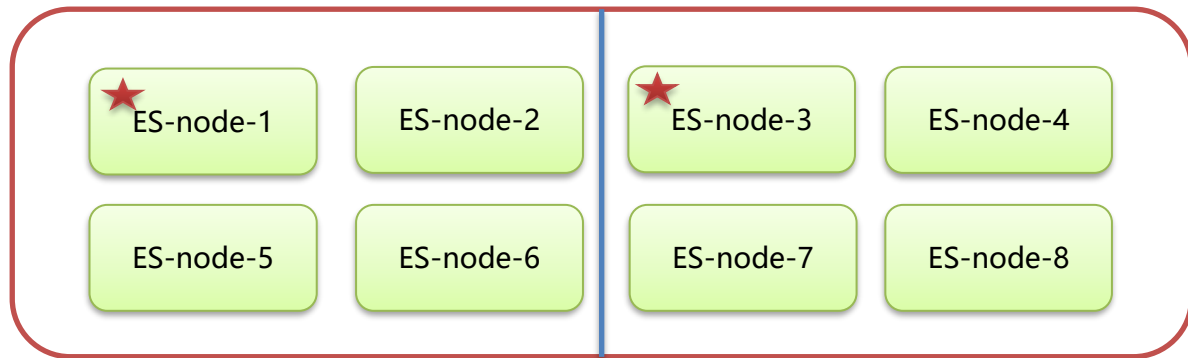
- 集群介绍
- 搭建集群
- Kibana 管理集群
- JavaAPI 访问集群
- 集群原理
- 脑裂
- 集群扩容

脑裂

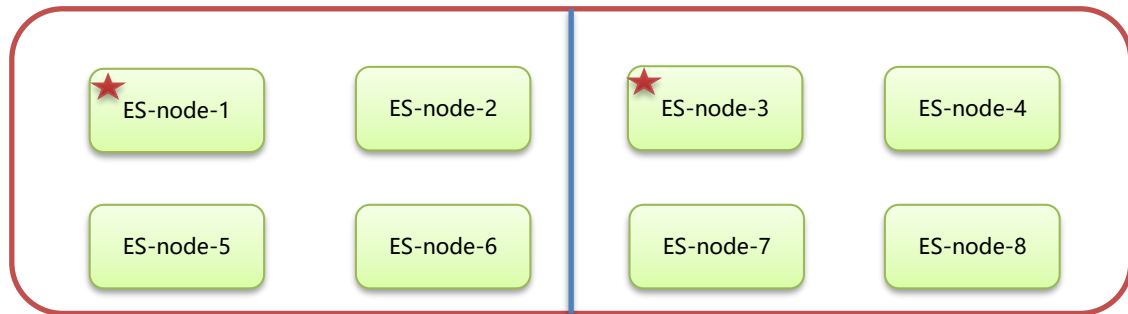


脑裂

- 一个正常es集群中只有一个主节点（Master），主节点负责管理整个集群。如创建或删除索引，跟踪哪些节点是集群的一部分，并决定哪些分片分配给相关的节点。
- 集群的所有节点都会选择同一个节点作为主节点。
- 脑裂问题的出现就是因为从节点在选择主节点上出现分歧导致一个集群出现多个主节点从而使集群分裂，使得集群处于异常状态。



脑裂原因



1. 网络原因：网络延迟

- 一般es集群会在内网部署，也可能在外网部署，比如阿里云。
- 内网一般不会出现此问题，外网的网络出现问题的可能性大些。

2. 节点负载

- 主节点的角色既为master又为data。数据访问量较大时，可能会导致Master节点停止响应（假死状态）。

```
#是不是有资格主节点
node.master: true
#是否存储数据
node.data: true
```

3. JVM内存回收

- 当Master节点设置的JVM内存较小时，引发JVM的大规模内存回收，造成ES进程失去响应。

避免脑裂

脑裂产生的原因：

1. 网络原因：网络延迟较高
2. 节点负载：主节点的角色既为master又为data
3. JVM内存回收：JVM内存设置太小

避免脑裂：

1. 网络原因：discovery.zen.ping.timeout 超时时间配置大一点。默认是3S
2. 节点负载：角色分离策略
 - 候选主节点配置为
 - node.master: true
 - node.data: false
 - 数据节点配置为
 - node.master: false
 - node.data: true
3. JVM内存回收：修改 config/jvm.options 文件的 -Xms 和 -Xmx 为服务器的内存一半。

ElasticSearch 集群管 理

- 集群介绍
- 搭建集群
- Kibana 管理集群
- JavaAPI 访问集群
- 集群原理
- 脑裂
- 集群扩容

集群扩容



传智播客旗下高端IT教育品牌