# 概述

## 概念

**索引**

一个索引是一系列有相同特征的文档的集合。

**类型（TYPE）--已弃用**

在一个索引中可以定义一个或多个类型，一个类型是对一个索引逻辑上的划分。比如在一个博客的索引中，可以有一个类型是用户数据，一个类型是博客数据，一个类型是评论数据。

不过在ES6.0版本之后，多类型映射已经不再支持，即同一个索引下仅可以存在一个类型。

在后期更高的版本中，类型参数可能会被完全废弃。

**文档**

文档是被索引的基本单元，ES不仅存储文档，而且索引每个文档的内容使之可以被检索。在ES中，你对文档进行索引、检索、排序和过滤。

ES使用JSON作为文档的序列化格式。JSON序列化被大多数编译语言所支持，并且已经成为NoSQL领域的标准格式。

它简单、简洁并且易于阅读。

**分片和副本（Shards/Replicas）**

一个索引可以存储很大的数据，这些空间可以超过一个节点的物理存储的限制。例如，十亿个文档占用磁盘空间为1TB。仅从单个节点搜索可能会很慢，还有一台物理机器也不一定能存储那么多的数据。为了解决这一问题，ES将索引分解成多个分片。当你创建一个索引，你可以简单地定义你想要的分片数量。每个分片本身是一个全功能的、独立的单元，可以托管在集群中的任何节点。

副本（Replicas），副本是一个非常有用的功能，不然会有单点问题。当网络中的某个节点出现问题的时候，复制可以对故障进行转移，保证系统的高可用。因此，ES允许你创建一个或多个拷贝，你的索引分片就形成了所谓的副本或副本分片。

## Solr

# 原理

## 存储管理

## 写入流程

## 读取流程

## Shard分配机制

# 使用

# 性能调优

## 写优化

### 批量提交

### 优化存储设备

### 合理使用归并

### 减少Refresh次数

### 加大Flush设置

### 减少副本数量

## 读优化

### 避免大结果和深翻

## Scroll查询

### Scroll-Scan查询

### 选择格式的路由

### SearchType

### 定期删除

## 堆大小设置

## 角色隔离和脑裂

### 角色隔离

### 避免脑裂

# 应用