

1、document 路由到 shard 上是什么意思

当客户端创建 document 的时候，es 此时就需要决定说，这个 document 是放在这个 index 的哪个 shard 上，这个过程就是 document 路由到 shard 上

2、路由算法：shard = hash (routing) % number_of_primary_shards

举个例子，一个 index 有 3 个 primary shard，P0，P1，P2

每次增删改查一个 document 的时候，都会带过来一个 routing number，默认就是这个 document 的 _id（可能是手动指定，也可能是自动生成），routing = _id，假设 _id = 1 会将这个 routing 值，传入一个 hash 函数中，产生一个 routing 值得 hash 值，

hash (routing) = 21，然后将 hash 函数产生的值就对这个 index 的 primary shard 的数量求余数 $21 \% 3 = 0$ ，就决定了，这个 document 就放在 P0 上。

这样就决定 document 在哪个 shard 上，最重要的一个值就是 routing 值，默认是 _id，也可以手动指定，相同的 routing 值，每次过来，从 hash 函数中，产生的 hash 值一定是相同的，无论 hash 值是几，无论什么数字，

对 number_of_primary_shards 求余数，结果一定是 $0 \sim \text{number_of_primary_shards} - 1$ 之间这个范围的 0，1，2。

hash(x)取余 ==> shard_num

3、_id or custom routing value(通过 id or 手动指定 来产生 routing)

默认的 routing 就是 _id

也可以在发送请求的时候，手动指定一个 routing value，比如说 `put /index/type/id?routing=user_id`

手动指定 routing value 是很有用的，可以保证说，某一类 document 一定被路由到一个 shard 上去，那么在后续进行应用级别的负载均衡，以及提升批量读取

4、primary shard 数量不可变的谜底

我们知道，一个index的数据会被分为多片，每片都在一个shard中。所以说，一个document，只能存在于一个shard中。

当客户端创建document的时候，es此时就需要决定说，这个document是放在这个index的哪个shard上。

这个过程，就称之为document routing，数据路由。

