ES

题目难度：★★★

知识点标签：ES

学习时长：20分钟

题目描述

ES 的数据读取过程以及文档读写原理大致分析一下?

写数据过程

1. 客户端通过hash选择一个node发送请求，这个node被称做coordinating node（协调节点），
2. 协调节点对docmount进行路由，将请求转发给到对应的primary shard
3. primary shard 处理请求，将数据同步到所有的replica shard
4. 此时协调节点，发现primary shard 和所有的replica shard都处理完之后，就反馈给客户端

写数据的底层原理：

1. 在到达primary shard的时候，数据先写入内存buffer， 此时，在buffer里的数据是不会被搜索到的同时生成一个translog日志文件， 将数据写入translog里
2. 如果内存buffer空间快man满了，就会将数据refresh到一个新的segment file文件中，而且es里每隔1s就会将buffer里的数据写入到一个新的segment file中，这个segment file就存储最最近1s中buffer写入的数据，如果buffer里面没有数据，就不会执行refresh操作，当建立segment file文件的时候，就同时建立好了倒排索引库。
3. 在buffer refresh到segment之前，会先进入到一个叫os cache中，只要被执行了refresh操作，就代表这个数据可以被搜索到了。数据被输入os cache中，buffer就会被清空了，所以为什么叫es是准实时的？NRT，near real-time，准实时。默认是每隔1秒refresh一次的，所以es是准实时的，因为写入的数据1秒之后才能被看到。还可以通过es的restful api或者java api，手动执行一次refresh操作，就是手动将buffer中的数据刷入os cache中，让数据立马就可以被搜索到。
4. 就这样新的数据不断进入buffer和translog，不断将buffer数据写入一个又一个新的segment file中去，每次refresh完buffer清空，translog保留。随着这个过程推进，translog会变得越来越大。当translog达到一定长度的时候，就会触发commit操作。translog也是先进入os cache中，然后每隔5s持久化到translog到磁盘中，
5. commit操作，第一步，就是将buffer中现有数据refresh到os cache中去，清空buffer 每隔30分钟flush
6. es也有可能会数据丢失，有5s的数据停留在buffer、translog os cache, segment file os cache中，有5s的数据不在磁盘上，如果此时宕机，这5s的数据就会丢失，如果项目要求比较高，不能丢失数据，就可以设置参数，每次写入一条数据写入buffer，同时写入translog磁盘文件中，但这样做会使es的性能降低。
7. 如果是删除操作，commit操作的时候就会生成一个.del文件，将这个document标识为deleted状态，在搜索的搜索的时候就不会被搜索到了。
8. 如果是更新操作，就是将原来的document标识为deleted状态，然后新写入一条数据
9. buffer每次refresh一次，就会产生一个segment file，所以默认情况下是1秒钟一个segment file，segment file会越来越多，当躲到一定程度的时候，es就会自动触发merge(合并)造作，将所有segment file文件 merge成一个segment file，并同时物理删除掉标识为deleted的doc，

es读取过程

1. 客户端发送get请求到任意一个node节点，然后这个节点就称为协调节点，
2. 协调节点对document进行路由，将请求转发到对应的node，此时会使用随机轮询算法，在primary shard 和replica shard中随机选择一个，让读取请求负载均衡，
3. 接收请求的node返回document给协调节点，
4. 协调节点，返回document给到客户端

文档读写模型实现原理

1. ElasticSearch，每个索引被分成多个分片（默认每个索引5个主分片primary shard），每个分片又可以有多个副本。当一个文档被添加或删除时（主分片中新增或删除），其对应的复制分片之间必须保持同步。那如何保持分片副本同步呢？这就是本篇重点要阐述的，即数据复制模型。

   ElasticSearch的数据复制模型是基于主从备份模型的。每一个复制组中会有一个主分片，其他分片均为复制分片。主分片服务器是所有索引操作的主要入口点（索引、更新、删除操作）。一旦一个索引操作被主服务器接受之后主分片服务器会将其数据复制到其他副本。

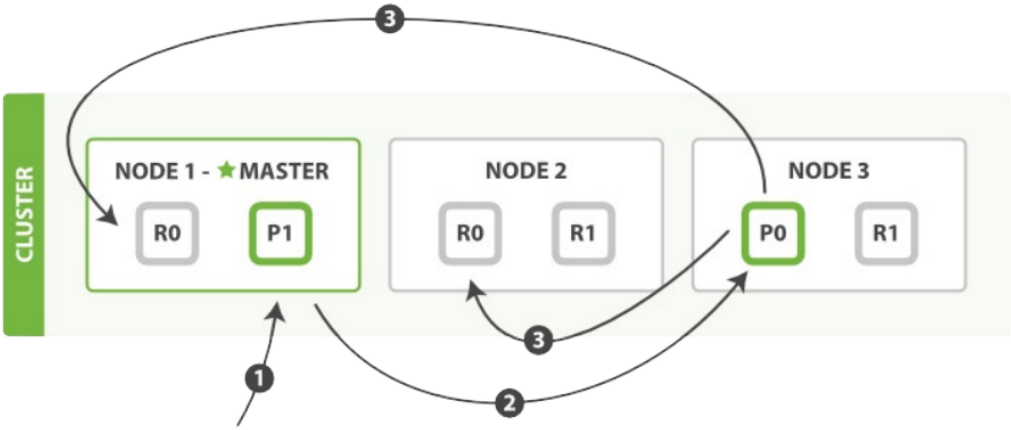
   2、基本写模型
   ElasticSearch每个索引操作首先会进行路由选择定位到一个复制组，默认基于文档ID(routing)，其基本算法为hash(routing) % (primary count)。一旦确定了复制组，则该操作将被转发到该组的主分片（primary shard）。主分片服务器负责验证操作并将其转发到其他副本。

   由于副本可以离线（一个复制组并不是要求所有复制分片都在线才能运作），可能不需要复制到所有副本。ElasticSearch会维护一个当前在线的副本服务器列表，这个列表被称为in-sync副本，由主节点维护。也就是当主分片接收一个文档后，需要将该文档复制到in-sync列表中的每一台服务器。

   主分片的处理流程如下：

   验证请求是否符合Elasticsearch的接口规范，如果不符合，直接拒绝。
   在主分片上执行操作(例如索引、更新或删除一个文档)。如果执行过程中出错，直接返回错误。
   将操作转发到当前同步副本集的每个副本。如果有多个副本，则并行执行。（in-sync当前可用、激活的副本）。
   一旦所有的副本成功地执行了操作并对主服务器进行了响应，主服务器向客户端返回成功。

   写请求的流程如下图所示（图片来源于《Elasticsearch权威指南》，如有侵权，马上删除）：



下面我们罗列在主分片和复制分片上成功新建、索引或删除一个文档必要的顺序步骤：

1. 客户端给 `Node 1` 发送新建、索引或删除请求。
2. 节点使用文档的 `_id` 确定文档属于分片 `0` 。它转发请求到 `Node 3` ，分片 `0` 位于这个节点上。
3. `Node 3` 在主分片上执行请求，如果成功，它转发请求到相应的位于 `Node 1` 和 `Node 2` 的复制节点上。当所有的复制节点报告成功， `Node 3` 报告成功到请求的节点，请求的节点再报告给客户端。

异常处理机制：

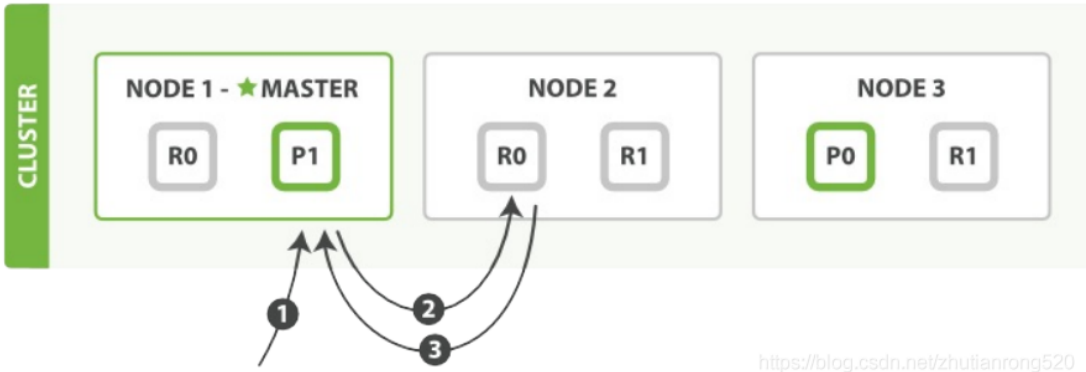
在索引过程中，许多情况会引发异常，例如磁盘可能会被破坏、节点之间网络彼此断开，或者一些配置错误可能导致一个副本的操作失败，尽管它在主服务器上是成功的。上述原因虽然很少会发生，但我们在设计时还是必须考虑如果发生错误了该如何处理。

另外一种情况异常情况也不得不考虑。如果主服务器不可用，ES集群该如何处理呢?

此时会触发复制组内的主服务器选举，选举后新的主节点会向master服务器发送一条消息，然后，该请求会将被转发到新的主服务器进行处理。此过程该请求在主节点选主期间会阻塞（等待），默认情况下最多等待1分钟。

注：主服务器(master)会监控各个节点的健康状况，并可能决定主动降级主节点。这通常发生在持有主节点的节点通过网络问题与集群隔离的情况下。

为了更好的理解master服务器与主分片所在服务器的关系，下面给出一个ElasticSearch的集群说明图：（图片来源于《Elasticsearch权威指南》，如有侵权，马上删除）



其中NODE1为整个集群的master服务器，而第一个复制组(P0,R0,RO,其主分片所在服务器NODE3)，第二个复制组(P1,R1,R1,其主分片所在服务器NODE1)。

一旦在主服务器上成功执行了操作，主服务器就必须确保数据最终一致，即使由于在副本上执行失败或由于网络问题导致操作无法到达副本（或阻止副本响应）造成的。

为了避免数据在复制组内数据的不一致性（例如在主分片中执行成功，但在其中一两个复制分片中执行失败），主分片如果未在指定时间内（默认一分钟）未收到复制分片的成功响应或是收到错误响应，主分片会向Master服务器发送一个请求，请求从同步副本中删除有问题的分片，最终当主分片服务器确向Master服务器的确认要删除有问题的副本时，Master会指示删除有问题的副本。同时，master还会指示另一个节点开始构建新的分片副本，以便将系统恢复到一个健康状态。

主分片将一个操作转发到副本时，首先会使用副本数来验证它仍然是活动的主节点。如果由于网络分区（或长GC）而被隔离，那么在意识到它已经被降级之前，它可能会继续处理传入的索引操作并转发到从服务器。来自陈旧的主服务器的操作将会被副本拒绝。当主接受到来自副本的响应为拒绝它的请求时，此时的主分片会向Master服务器发送请求，最终将知道它已经被替换了，后续操作将会路由到新的主分片服务器上。

如果没有副本，那会发生什么呢?

这是一个有效的场景，可能由于配置而发生，或者是因为所有的副本都失败了。在这种情况下，主分片要在没有任何外部验证的情况下处理操作，这可能看起来有问题。另一方面，主分片服务器不能自己失败其他的分片（副本），而是请求master服务器代表它这样做。这意味着master服务器知道主分片是该复制组唯一的可用拷贝。因此，我们保证master不会将任何其他（过时的）分片副本提升为一个新的主分片，并且任何索引到主分片服务器的操作都不会丢失。当然这样意味着我们只使用单一的数据副本，物理硬件问题可能导致数据丢失。请参阅Wait For Active Shards，以获得一些缓解选项,(该参数项将在下一节中详细描述)。

注：在一个ElasticSearch集群中，存在两个维度的选主。Master节点的选主、各个复制组主分片的选主。

3、基本读模型

在Elasticsearch中，可以通过ID进行非常轻量级的查找，也可以使用复杂的聚合来获取非平凡的CPU能力。主备份模型的优点之一是它使所有的分片副本保持相同（除了异常情况恢复中）。通常，一个副本就足以满足读取请求。

当一个节点接收到read请求时，该节点根据路由规则负责将其转发给相应的数据节点，对响应进行整理，并对客户端作出响应。我们称该节点为该请求的协调节点。基本流程如下：

将读请求路由到到相关的分片节点。注意，由于大多数搜索条件中不包含分片字段，所以它们通常需要从多个分片组中读取数据，每个分片代表一个不同的数据子集（默认5个数据子集，因为ElasticSearch默认的主分片个数为5个）。
从每个分片复制组中选择一个副本。读请求可以是复制组中的主分片，也可以是其副本分片。在默认情况下，ElasticSearch分片组内的读请求负载算法为轮询。
根据第二步选择的各个分片，向选中的分片发送请求。
汇聚各个分片节点返回的数据，然后返回个客户端，注意，如果带有分片字段的查询，将之后转发给一个节点，该步骤可省略。
异常处理：
当一个碎片不能响应一个read请求时，协调节点将从同一个复制组中选择另一个副本，并向其发送查询请求。重复的失败会导致没有分片副本可用。在某些情况下，比如搜索，ElasticSearch会更倾向于快速响应（失败后不重试），返回成功的分片数据给客户端，并在响应包中指明哪些分片节点发生了错误。

4、Elasticsearch主备模型隐含含义

在正常操作下，每个读取操作一次为每个相关的复制组执行一次。只有在失败条件下，同一个复制组的多个副本执行相同的搜索。
由于数据首先是在主分片上进行索引后，然后才转发请求到副本，在转发之前数据已经在主分片上发生了变化，所以在并发读时，如果读请求被转发到主分片节点上，那该数据在它被确认之前（主分片再等待所有副本全部执行成功）就已经看到了变化。【有点类似于数据库的读未提交】。
主备模型的容错能力为两个分片（1主分片，1副本）。、

5、ElasticSearch 读写模型异常时可造成的影响

在失败的情况下，以下是可能的：

1个分片节点可能减慢整个集群的索引性能
因为在每次操作期间(索引)，主分片在本地成功执行索引动作后，会转发请求到期复制分片节点上，此时主分片需要等待所有同步副本节点的响应，单个慢分片可以减慢整个复制组的速度。当然，一个缓慢的分片也会减慢那些被路由到它的搜索。
脏读
一个孤立的主服务器可以公开不被承认的写入。这是由于一个孤立的主节点只会意识到它在向副本发送请求或向主人发送请求时被隔离。在这一点上，操作已经被索引到主节点，并且可以通过并发读取读取。Elasticsearch可以通过在每秒钟（默认情况下）对master进行ping来减少这种风险，并且如果没有已知的主节点，则拒绝索引操作。
本文详细介绍了ElasticSearch文档的读写模型的设计思路，涉及到写模型及其异常处理、读模型及其异常处理、主备负载模型背后隐含的设计缺陷与ElasticSearch在异常情况带来的影响。

es搜索过程

1. 客户端发送请求到协调节点，
2. 协调节点将请求大宋到所有的shard对应的primary shard或replica shard；
3. 每个shard将自己搜索到的结果返回给协调节点，返回的结果是dou.id或者自己自定义id，然后协调节点对数据进行合并排序操作，最终得到结果。
4. 最后协调节点根据id到个shard上拉取实际 的document数据，左后返回给客户端。