最近，一位之前一直在寻找它的用户在边肖向我们提出了一个问题。相信这也是很多币圈朋友经常疑惑的问题：elasticsearch架构相关问题，elasticsearch数据结构设计相关问题，带着这个问题，让专业的编辑来告诉你为什么。

分布式搜索引擎：将大量的索引数据分成多个块，每台机器放一部分，然后用多台机器搜索分散的数据，所有操作分布在多台机器上。，形成完整的分布式架构。

近实时，它有两层含义：

集群包含多个节点，每个节点属于哪个集群由一个配置决定，

节点是集群中的一个节点。节点也有一个名称，默认情况下是随机分配的。默认节点将加入一个名为elasticsearch的集群。如果你直接启动一堆节点，它们会自动形成一个elasticsearch集群。当然，一个节点也可以组成一个elasticsearch集群。

单据是es中最小的数据单位，一个单据可以是一条客户数据、一条商品分类数据、一条订单数据。，通常用json数据结构表示。索引下的每种类型可以存储多个文档。

文档中有多个字段。，每个字段是1个数据字段。

ES中多个节点集群时，会自动选举一个节点为主节点，实际上是在做一些管理工作。，比如维护索引元数据，负责切换主分片和副本分片的身份。如果主节点关闭，一个节点将被重新选为主节点。。如果非主节点发生故障，主节点会将该故障节点上的主碎片的身份转移到其他机器上的副本碎片。如果你修好了那台机器，重启后，主节点将控制缺失副本分片的分配，同步后续修改的数据，使集群恢复正常。更简单地说，如果一个非主节点发生故障，，那么这个节点上的主分片就没有了。好的，master会将主碎片对应的副本碎片(在其他机器上)切换到主碎片。。如果当机修复，修复后的节点不再是主分片，而是副本分片。

索引可以分成多个分片，每个分片存储一部分数据。。分割多个碎片是有益的。首先，它支持横向扩展。例如，如果您的数据量为3T，有三个碎片，每个碎片将有1T的数据。如果现在数据量增加到4T，怎么扩展就很简单了。，用四个shard重建一个索引，导入数据；二是提高性能。数据分布在多个shard上，也就是多台服务器上，所有的操作都会并行执行，分布在多台机器上，从而提高吞吐量和性能。。然后这个分片的数据其实有多个备份，也就是说每个分片都有一个主分片，负责写数据，但是也有多个副本分片。。主碎片写入数据后，会将数据同步到其他几个副本碎片。

使用这种复制方案，每个碎片&#039；的数据有多个备份。如果一台机器坏了，它不会。没关系，在其他机器上有其他数据副本，所以它是高度可用的。

总结：分发意味着两点：1。通过分片切片进行水平扩展；2.高可用性是通过副本机制实现的。

基本概念

数据写入过程：客户端通过hash选择一个节点发送请求，这个节点称为协调节点，协调节点路由docmount。，将请求转发到对应的主分片，主分片处理请求并将数据同步到所有副本分片。这时，协调节点。在发现主分片和所有副本分片都被处理后，会反馈给客户端。

客户端向任意一个节点发送get请求，然后这个节点称为协调节点。协调节点路由该文档并将请求转发给相应的节点。此时，使用随机轮询算法来随机选择主碎片和副本碎片中的一个，从而平衡读取请求的负载。接收请求的节点将文档返回到协调节点，协调节点，并将文档返回到客户端

es。最厉害的是做全文检索，就是比如你有三条数据

。1.java真的很有趣。

2.java好难学

3.J2EE特别牛逼

你根据java关键词搜索，搜索出包含java的文档。

更新/删除数据的过程首先是写和合并操作，然后在刷新过程中：

1。写过程与上面一致；

2。刷新过程有点不一样

所谓倒排索引。，就是先把你的数据内容分成单词，把每个句子分成一个个的关键词，然后记录下每个关键词出现在哪些id标注的数据中。

然后你可以根据这个id从其他地方找到相应的数据。这就是倒排索引的数据格式和搜索方法，也叫全文检索。

倒排索引是我们常见的倒排索引，主要包括两部分：

。

有序数据字典(包括单词$Term及其出现频率)。

对应于单词$Term的发布(即带有该单词的文件)

我们在搜索的时候，首先对搜索的内容进行分解，然后在字典中找到对应的$Term，从而找到与搜索相关的文件内容。

本质上存储字段是一个简单的键值对。默认情况下，StoredFields为false，Elastic Search存储整个文件的JSON源代码。

在什么情况下需要显式指定store属性？大多数情况下是不必要的。从_source获取值快速高效。如果您的文档很长存储_source或从_source获取字段的开销非常大。您可以将某些字段的store属性显式设置为yes。如上所述的缺点：假设你保存了10个字段。，而如果要获取这10个字段的值，需要多个io，如果要从Storedfield获取，只需要一个，而且_source是压缩的。

此时可以将某些字段指定为true，这意味着该字段的数据将被分开存储(实际上有两份，源和存储字段各一份)。此时此刻如果请求返回field1(store:yes)，es会识别出field1已经被存储，所以不会从_source加载，而是从field1的存储块加载。

Doc_values本质上是一种序列化的列存储，这种结构非常适合聚合、排序、脚本访问字段等操作。。而且这种存储方式对于压缩也非常方便，尤其是对于数字类型。这样可以减少磁盘空间，提高访问速度。ElasticSearch可以将索引中的所有文档值读入内存进行操作。

Doc_values存在于磁盘上

在es中，文本类型字段默认只会创建倒排索引，其他类型在创建倒排索引时也会创建正排索引。当然，es支持定制。。在这里，这个正交索引实际上是Doc值。

即上述动态索引

写入es的数据实际上是写入磁盘文件。查询时，，操作系统会自动将磁盘文件中的数据缓存到文件系统缓存中。

es的搜索引擎非常依赖底层文件系统缓存。如果你给文件系统缓存更多的内存，并且尽量让内存能够容纳所有的idx段文件索引数据文件，那么你在搜索的时候基本都会占用内存，性能会很高。。性能差距能有多大？我们之前的很多测试和压力测试，如果拿盘的话肯定是秒级的，搜索性能绝对是秒级的，1秒，5秒，10秒。但是如果你使用文件系统缓存，你使用的是纯内存。一般来说性能比磁盘高一个数量级，基本都是毫秒级，从几毫秒到几百毫秒不等。

那么我们如何节省文件系统缓存的空间呢？

向ES写入数据时，要考虑数据的最小化。当一行数据超过30个字段时，没有必要将所有数据都写入ES，只需检索需要的键列即可。可以缓存的数据越多。。因此，最好将每台机器写入的数据控制在小于或等于或略大于文件系统缓存空间。如果要搜索海量数据，可以考虑使用ESHbase架构。使用Hbase存储海量数据，然后ES搜索单据id，再去Hbase根据单据id查询指定的行数据。

当每台机器写入的数据比缓存os大太多时，太多的数据无法放入缓存。，那么就可以把一些热数据刷到缓存里了。

对于那些你认为比较热，经常访问的数据，最好做一个专门的缓存预热系统，也就是每隔一段时间就提前访问热数据。，让数据进入文件系统缓存。所以下次有人来访，表现肯定会好很多。

将热数据和冷数据分开，写入不同的索引，然后一定要将热索引数据刷到缓存中。

在ES中，最好不要使用复杂的关联表操作。当需要这样的场景时，可以在创建索引时关联数据。。比如在mysql中，需要根据相关ID查询两个表的相关数据：selecta.name，b.agefromajoinb其中a.id=b.id。写ES的时候，直接把关联的数据放在一个文档里就行了。

es分页被坑，为什么？例如，如果每页有10条数据，现在就要查询第100页。事实上，存储在每个碎片上的前1000条数据将在一个协调节点上找到。如果你有五个分片，那么就会有5000条数据，然后协调节点会合并处理这5000条数据。，然后得到第100页的最后10条数据。

分布式，你要检查100页的10条数据。不可能说从五个分片中，每个分片会检查2条数据。，最后在协调节点合并成10条数据？你要从每个分片中查找1000条数据，然后根据你的需要进行排序、筛选等等，最后再次分页得到第100页的数据。当你翻页的时候转得越深，每个分片返回的数据越多，协调节点处理的时间越长，非常尴尬。所以你用es做分页的时候会发现越往后翻越慢。

我们以前遇到过这个问题。以es为分页，前几页需要几十毫秒，翻到10页或者几十页，基本上5~10秒就能找出一页数据。

解决方案？

1)不允许深度分页：请与产品经理联系。你的系统不允许翻这么深的一页。默认情况下，转得越深，性能越差。

2)在APP或微信官方账号中，通过下拉实现分页，即下拉时获取最新页面，可通过scrollapi实现；

scroll会为你一次生成所有数据的快照，然后每次滑回page，通过移动光标scroll_id得到下一页，性能会比上面说的分页性能高很多，基本以毫秒为单位。。但是只有1的缺点是适合类似微博的场景&#039；下拉翻页，就可以&#039；不要随意跳到任何一页。换句话说，你可以&#039；不要跳到第10页，然后跳到第120页，再回到第58页。你可以&#039；不要随意在页面上跳来跳去。。所以现在很多APP产品不&#039；t不允许你随意翻页，也有一些网站你只能下拉一页一页的翻。

必须在初始化期间指定scroll参数。，告诉es保存该搜索上下文的时间。你需要确保用户不会&#039；不要连续几个小时翻页，否则他们可能会因为超时而失败。

除了使用滚动api，也可以使用search_after。search_after的思路是利用上一页的结果来帮助检索下一页的数据。很明显，这种方法不允许你随意翻页，只能一页一页的往后翻。初始化时，您需要使用一个只有1的字段作为排序字段。

在你了解ES之前，你要了解他的底新月。请参考我的文章

lunce入门—

新月体写作流程

首先，新月体&#039；的写数据流是每一段数据先写入缓冲区再生成一个段，每个段文件都刷新到硬件系统。。当buffer被写入文件系统生成段时，会被转化为translog文件，以保证文件写入的成功。如果失败，将通过translog文件重写。。每个段文件都将记录在提交文件中。至于什么时候写到本地文件，取决于segs的数量或者translog的大小。。每次索引、批量、删除和更新完成时，它必须触发刷新到磁盘的事务日志。如果不保证通过可行性来提高性能，可以设置

{

。

？"index.translog。耐久性"&quot异步&quot

}

弹性搜索写入流程

对于ES来说，首先是集群，而不是单机。此时，索引是面向集群的。而单机的索引就是当前的分片。众所周知，每个缓冲区都是一个段，它会影响存储和电缆施工的性能。对于ES来说，就是合并段和合并操作。可以根据段的大小和数量来合并策略。当然，合并次数也会降低系统的性能，所以可以通过forcemerge接口人为设置合并次数。

Whenelasticsearchconfirmsthestoragerulesofdatalocation

？

shard=hash(routing)%number_of_primary_shards

并且每条数据都有一个id，然后数据的存储位置由切片数决定。

那么在设计集群的时候就会有另外一个概念，那就是复制，保证数据的一致性。,whichcanbeunderstoodasnamenodeandsecondnamenodeinhadoop.

ESparameterdetailedcontrol

cluster,routing,allocationandenabling.

该参数用于控制允许分配哪个片。默认为全部。选项还包括primaries和new_primaries。没有一个完全拒绝碎片化。该参数的功能将在本书后面的集群升级一章中解释。

星团。路由。分配。允许_重新平衡

该参数用来控制什么时候允许数据均衡。默认是indices_all_active即要求所有段都能够成功启动，才能进行数据均衡操作，否则在集群重启阶段会浪费太多的流量。

cluster.routing.allocation.cluster_concurrent_rebalance

该参数用于控制集群中同时运行的数据平衡任务的数量。默认值为2。。如果有节点增减，集群的负载压力不高，可以适当增加。

cluster.routing.allocation.node_initial_primaries_recoveries

该参数用于控制节点重启时，允许同时恢复几个主片。默认值为四。如果节点是多磁盘，IO压力不大，可以适当增加。

集群。路由。分配。node_concurrent_recoveries

该参数用于在除主分区重启和恢复以外的其他情况下控制节点。，允许同时执行数据恢复任务。默认值为2。因此，当节点重启时，可以看到主碎片的恢复完成得很快，而副本碎片的恢复非常慢。除了副本碎片本身的数据通过网络被复制，并发线程本身也减少了一半。当然这个设定也是合理的。——主切片必须本地恢复，但副本切片需要经过网络，带宽有限。从ES1.6开始，可以本地恢复冷索引的副本碎片，可以适当增加这个参数。

指示器。恢复。concurrent_streams

该参数用于控制节点从网络复制和恢复副本碎片时的数据流数量。默认值为三。它可以与以前的配置一起增加。

指示器。recovery.max_bytes_per_sec

该参数用于控制节点恢复的速率。默认值为40MB。很明显是比较小的，建议加大。

也可以通过重新路由界面手动控制碎片。

弹性搜索中冷热数据读写分离

弹性搜索集群的一个突出问题是，当用户进行大查询时，大量的IO读取和聚集计算会导致机器负载和CPU利用率的增加，从而影响新数据的写入，这个过程甚至会持续几分钟。。所以可能需要像MySQLcluster一样做读写分离。

实施例

N台机器用来存储热力数据，上面只放当天的数据。。节点之前的数据。attr.tag:hot

配置在elasticsearc.yml上面这N个热数据节点放在另外M台机器上。.Themcoldda tanodesareconfiguredwithnode.attr.tag:stall

elasticsearcharchitectureprincipleghoststory-CSDNblog

.

Elasticsearch是由ShayBanon发起的开源搜索服务器项目，于2010年2月发布。到目前为止，该项目已经发展成为搜索和数据分析解决方案领域的主要参与者。，广泛应用于知名或鲜为人知的搜索应用。此外，由于它的分布式和实时功能，许多人把它当作一个文档数据库。

Elasticsearch架构简单介绍如下。

索引

索引是Elasticsearch对逻辑数据的逻辑存储，所以可以分成更小的部分。你可以把索引想象成一个关系数据库的表。但是，索引的结构是为快速有效的全文索引准备的，尤其是它不存储原始值。。如果你了解MongoDB，你可以把Elasticsearch的索引看成是MongoDB中的一个集合。如果您熟悉CouchDB，您可以将该索引视为CouchDB数据库索引。。Elasticsearch可以在一台机器或多台服务器上存储索引。每个索引有一个或多个碎片，每个碎片可以有多个副本。

文档

elasticsearch中存储的主要实体称为文档。用关系数据库做类比，文档相当于数据库表中的一行记录。。在对比Elasticsearch和MongoDB中的文档时，你会发现两者可以有不同的结构，但是Elasticsearch文档中相同的字段必须有相同的类型。这意味着所有包含标题字段的文档，标题字段类型必须相同，如字符串。

一个文档由多个字段组成，每个字段可能在一个文档中出现多次。这种字段称为多值字段。每个字段都有一个类型。，如文本、值、日期等。字段类型也可以是复杂类型，并且字段包含其他子文档或数组。字段类型在Elasticsearch中非常重要，因为它们提供了有关如何执行各种操作(如分析或排序)的信息。幸运地这可以自动确定，但是，我们仍然建议使用映射。与关系数据库不同，文档不需要有固定的结构，每个文档可以有不同的字段。此外，在程序开发期间，没有必要确定哪些字段是可用的。当然，您可以使用模式来强制文档结构。。从客户端&#039；从的角度来看，文档是一个JSON对象(有关JSON格式的更多信息，请参见)。每个文档都存储在一个索引中，并有一个唯一的标识符和由Elasticsearch自动生成的文档类型。。

文档需要有对应于文档类型的惟一标识符，这意味着两种不同类型的文档在一个索引中可以有相同的惟一标识符。

文档类型

在Elasticsearch中。索引对象可以存储许多用于不同目的的对象。例如，博客应用程序可以保存文章和评论。文档类型允许我们在单个索引中轻松区分不同的对象。每个文档可以有不同的结构，但是在实际部署中按类型区分文件对数据操作很有帮助。当然，需要记住的是，不同的文档类型不能为同一个属性设置不同的类型。例如，在同一索引中的所有文档类型中，名为title的字段必须具有相同的类型。

映射

在上述文章内容为全文检索基础知识的部分，我们提到了分析过程：准备输入文本进行索引和检索。文档中的每个字段都必须根据不同的类型进行分析。例如比如前者的数字不要按字母顺序排序，后者的第一步是忽略HTML标签，因为它们是无用的信息噪音。。Elasticsearch将上述文章内容存储在地图中作为该字段的信息。每种文档类型都有自己的映射，即使我们没有&#039；Idon’我无法清楚地定义它。

现在我们已经知道，Elasticsearch将数据存储在一个或多个索引中，每个索引包含各种类型的文档。我们还知道每个文档有许多字段，映射定义了Elasticsearch如何处理这些字段。但是还有更多。从一开始，Elasticsearch就被设计为一个分布式解决方案，每秒钟可以处理数亿个文档和数百个搜索请求。这是由于几个重要的概念，我们现在将更详细地描述这些概念。

节点和群集

Elasticsearch可以作为独立的单一搜索服务器使用。但是，为了能够处理大型数据集并实现容错和高可用性，Elasticsearch可以运行在许多合作服务器上。。这些服务器称为集群，组成集群的每个服务器称为一个节点。

碎片化

当有大量文档时，由于内存的限制，硬盘容量和处理能力不足，无法足够快地响应客户端请求。，一个节点可能不够。在这种情况下，数据可以分成更小的部分，称为碎片(每个碎片是一个独立的ApacheLucene索引)。每个切片可以放在不同的服务器上，所以，数据可以在集群的节点中传播。当查询的索引分布在多个片上时，Elasticsearch会将查询发送给每个相关的片，并将结果合并在一起，但应用程序不

会&#039；我不知道切片的存在。此外，多个切片可以加快索引速度。

副本

为了提高查询吞吐量或实现高可用性，可以使用碎片副本。副本只是切片的精确副本，每个切片可以有零个或多个副本。换句话说Elasticsearch可以有许多相同的切片，自动选择其中一个切片来更改索引操作。这个特殊碎片叫做主碎片，其他的叫做副本碎片。。当主切片丢失时，例如，切片数据所在的服务器不可用，群集会将副本提升到新的主切片。

？1.索引包含多个碎片。

？2.每个碎片是最小的工作单元，它承载部分数据、lucene实例以及建立索引和处理请求的完整能力。。添加或删除节点时，shard会自动在节点中进行负载均衡。

？3.主碎片和副本碎片，每个文档必须只存在于一个主碎片及其对应的。在副本碎片中不可能存在多个主碎片。

？4.副本碎片是主碎片的副本，负责容错和读取请求负载。副本中的数据保证是强一致或最终一致的。

？5.主碎片的数量在创建索引时是固定的，因为在索引时，文档需要根据主碎片的数量进行路由(默认情况下，文档&#039；s_id属性用于获取路由的哈希值。，或者您可以使用路由来指定其他文档字段采用哈希值进行路由)。副本分片的数量可以随时修改。

？6.主碎片的默认数量是5。Bydefault,thereplicais1,andthereare10fragments,5masterfragmentsand5replicafragmentsbydefault.

？7.主分片不能和自己的副本分片放在同一个节点上(否则如果节点宕机，主分片和副本都会丢失，不能起到容错的作用)。，但它可以与其他主碎片的副本碎片放在同一节点上。

都看过了吗？我相信你现在对elasticsearch架构已经有了初步的了解。！也可以收藏页面获取更多关于elasticsearch数据结构设计的知识！区块链，虚拟货币，我们是认真的！