ES手册

本手册将介绍如何搭建一个适当的ES集群，来满足业务需求，对于ES的关键性能提供定性的分析介绍。

1、针对业务选择合适的集群

ES到底都能干神马用？ES在研发之初只想做全文检索，而且对于英文分词的支持非常好，但后来人们发现，这个玩意部署起来实在是太方便了，而且很容易横向扩展，于是开始在上面进行各种尝试。

1.1 离线数据分析

最常用的其实是日志分析，实时数据统计分析也算日志分析的一部分。著名的ELK就是一套成熟的日志分析与手机的解决方案。其中的Logstash就是用来把日志分割成单独的字段，并且保存在ES中。

这种服务的特点是，数据写入量大，查询QPS相对较小（用作分析工具），但可能有单个查询耗时很长的大查询（BigQuery），这种查询可能导致集群宕机，但这些功能都不用保证完全高可用，只要能保证数据能够及时写入，出现宕机及时恢复，有数据补偿即可。

服务特点：

项目	具体需求	注意事项
CPU	常规	
memory	主工作集群需要高内存，推荐64GB	
disk	如果数据写入量大，主工作集群需要挂载多物理硬盘或其他优化的磁盘写入方案	注意bulk和普通方式存储并不存在本质区别，如果普通index出现瓶颈
network	无特殊需求	
查询qps	10以下	
index qps	200以上	
BigQuery数据大小	聚合千万级以上别数据，返回在10万行以上	
集群高可用	否	
集群状态监控	是	实时获取集群状态，宕机时报警
集群宕机及时恢复	是	需手动及时恢复
主备集群	否	
集群宕机启用备用集群	否	
数据补偿	是	

1.2 排序或者索引服务

ES还经常被用作高性能的实时排序或者索引服务。比如朋友圈或者知乎的Timeline：

这个功能用ES来实现其实非常简单。如果将在Timeline上出现的每个条目定义为Item，则在ES中只需存储ItemID，CreateTime两个字段，就能完成基本的Timeline功能。使用ES的查询排序功能，可以快速生成一张按时间排序的ID列表。结合滚动查询功能，可以方便的查看整个排序列表直到最早的一条。

再有就是用作联想数据框的提示：

这个功能也非常简单，通常需要用一个合适的分词器，将中文按设置好的规则分词，通过输入内容匹配ES中的字符串，拉出最符合的N个项即可，为了提高查询速度，ES中甚至不需要存储Item的ID，只需要中文名即可。

这类服务的特点是，数据写入量不会太大，由于只保存索引数据，通常写入量很小；查询QPS高，可以说是整个社交类App最主要的查询接口，但查询的结果通常不会很大；通常这种服务都要求高可用。

服务特点：

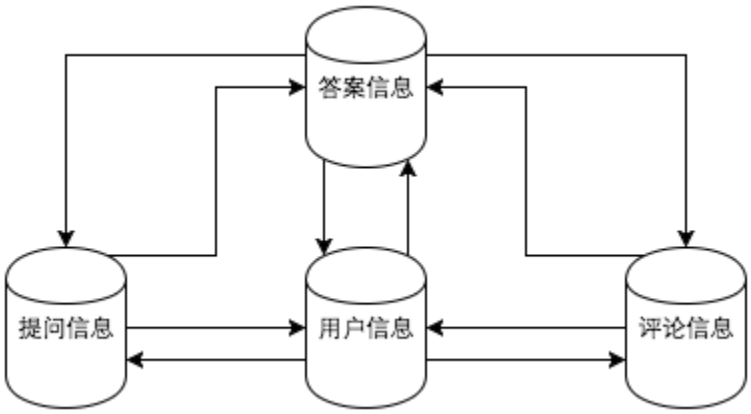
项目	具体需求	注意事项
CPU	尽可能高	排序服务是一项CPU密集的
memory	主工作集群内存可以32GB以内，但不能低于16GB	查询量大，但单个查询较小，而且可以应用ES缓存，整体对内存需求
disk	无特殊要求，能存储下数据即可	
network	无特殊需求	
查询qps	1000以上	
index qps	非常小	
BigQuery数据大小	无	
集群高可用	是	
集群状态监控	是	实时获取集群状态，宕机时报警
集群宕机及时恢复	否	有备用集群，可以不及时恢复
主备集群	是	
集群宕机启用备用集群	是	
数据补偿	否	

1.3 实时搜索（当做DB）未完待续...

将ES用作主要业务数据服务，目前还没有听说有哪家完全脱离数据库，单独使用ES当主要数据服务。这里只是简要分析下这种服务的可能性，具体实践还请大家勇敢尝试！

(1)典型的UGCApp分析（知乎），业务比较简单：用户数据，问题数据，答案数据，评论数据

数据抽象：



几个典型的业务场景和注意事项：

一个用户搜索一个关键字，查找出了N个问题下的答案

一个用户在一个问题下面添加了一个答案

一个用户评论了一个答案

一个用户关注了另一个用户

2、三种集群搭建举例

节点类型：

虽然ES官方说自己高可用，但其实可用性并没有那么高，一个合理的集群结构能提高ES的可用性，毕竟高可用是个很复杂的工作，从50%提高到99%可能并不难，而后面的0.99%才是最复杂最困难的。

所以我根据官方文档和自己的经验，将ES集群中的机器分成一下几种类型。

节点类型	节点作用
ControlNode(MasterNode)	当做主节点，不保存数据，也不提供数据查询服务，只用于维护集群状态
ClientNode	负载均衡节点，不保存数据，但提供数据查询服务(只接收和转发请求)
DataNode	数据存储和运算节点，保存数据，不直接提供查询服务（接收Client的查询请求）
BackupNode	数据备份节点，跟DataNode有相同的数量，实时备份DateNode数据，当DataNode宕机，由ControlNode发现
TribeNode	集群组节点，可以连接多个集群，进行写入和读取操作

这些节点各有各的作用，在不同的集群中，可能并不需要所有类型的节点，但根据业务不同，合理的选择功能不同的节点，提高整个集群的性能或者稳定性。

接下来将提供4个集群搭建的例子，分别适应不同的业务需求，当然集群管理的大部分功能都需要我们开发。

2.0 懒人集群

一种最普通的集群，可以认为是一种通用集群，不具备集群的高可用，但普通的业务需求是可以满足的，目前订单所用的集群就是这种。

节点类型	机器数量
ControlNode	0
ClientNode	0
DataNode	N(同时也是ControlNode，都可以被选举为Master)
BackupNode	0
TribeNode	0

集群特点：容易搭建，运维简单，因为所有机器配置都一样，有问题，ssh脚本重启即可。但显然当集群有机器挂掉时没有备份方案，只能依靠ES自身的高可用，但如果分片设置不合理，可能挂掉一台，就没法用了。重点是，挂掉后恢复很慢。

2.1 离线数据分析集群

参考1.1节的介绍，这种集群需要如下配置的机器：

节点类型	机器数量	用途
ControlNode	2~3	1.保证集群任何时刻都有Master节点 2.保证集群中每台机器的状态都可见 3.控制集群启动，停止，增加、减少节点等操作
ClientNode	3	1.负载均衡，接收请求，并转发给DataNode处理 2.数据写入失败需要记录，待集群恢复后重新写入
DataNode	N(根据数据量大小)	1.主工作机，执行查询请求。
BackupNode	0	
TribeNode	0	

注意：ControlNode的用途1~3，ES并没有相应的实现，需要自己实现。这种集群可以提供快速的实时数据分析功能，同时提供较高的写入速度，但查询并发能力并不高。

需要自己实现的部分：集群中ClientNode和DataNode的状态监控，数据写入失败的记录，以及集群恢复时重新写入失败数据。

2.2 排序或索引集群

参考1.2节介绍，集群可配置如下：

节点类型	机器数量	用途
ControlNode	2~3	1.保证集群任何时刻都有Master节点 2.保证集群中每台机器的状态都可见 3.控制集群启动，停止，增加、减少节点等操作 4.发现掉线集群，立刻通知BackupNode
ClientNode	5	1.负载均衡，接收请求，并转发给DataNode处理
DataNode	N(根据数据量大小)	1.主工作机，执行查询请求。
BackupNode	N	1.当ControlNode发现有DataNode下线，立刻加入BackupNode
TribeNode	0	

注意：BackupNode需要自己维护，跟DataNode保持强一致，这点可通过rsync服务解决。

2.3 实时搜索集群（当做DB）

参考1.3节介绍，集群可配置：

节点类型	机器数量	用途
ControlNode	2~3	1.保证集群任何时刻都有Master节点 2.保证集群中每台机器的状态都可见 3.控制集群启动，停止，增加、减少节点等操作
ClientNode	3	1.负载均衡，接收请求，并转发给DataNode处理 2.数据写入失败需要记录，待集群恢复后重新写入 3.配置路由规则，将不同热度的数据写入不同的集群中 4.注意配置机器半连接队列，timewait快速回收，提高系统的连接数
DataNode	N(根据数据量大小)	1.主工作机，执行查询请求。 2.分配热查询集群
BackupNode	N+1	1.热查询集群备份 2.ClientNode备份
TribeNode	0	

注意：这类集群需要热数据的路由，将性能最好的机器集中到热查询上来，当然，最好是机器都很好。配置系统参数可以参考Linux相关数据，主要原则是提高连接数，因为如果后端机器返回慢，可能会导致ClientNode主动关闭连接，而导致timewait，这种情况机器很难恢复。

3、集群运维

首先我们要明确ES集群在啥情况下会挂。

1)疯狂写入，磁盘又没有优化。这种情况的初期是有大量写入错误日志，写入性能下降，正常index一般7-8ms，一般这种情况index会超过100ms，这时如果还不停止写入，那么ES很快会挂掉。

2)执行一个超大的聚合查询，返回结果在百万行级别，而你的DataNode实例却只有十几GB的内存，那么这时会有几台机器内存急剧升高，最终O

OM，这时ES无法完成任何相应。

3)机器不稳定，导致但节点进程意外终止。

其次，这些情况的挂，会产生啥影响？

1)挂的彻底，如果机器配置不均衡，可能会产生雪崩效应，一台接一台的挂。写入和查询都不能用。

2)挂的不彻底，集群响应炒鸡慢，查询相应会稍好于写入，及时停止查询请求，可能能时集群恢复，但一般情况是不能恢复的。

3)集群没有挂，查询还是可以使用，但这是集群的网络会比较高，因为需要复制出那些挂掉的分片，查询可用，写入可能会出现不一致(主备分片数据不一致)。

最后，我们如何手动解决这些问题

1)只能停写，加机器，重启。不要抱有幻想 ，磁盘写入性能不够，从ES层面是没法解决的，要么挂多硬盘，要么换SSD，要么配置RAID。

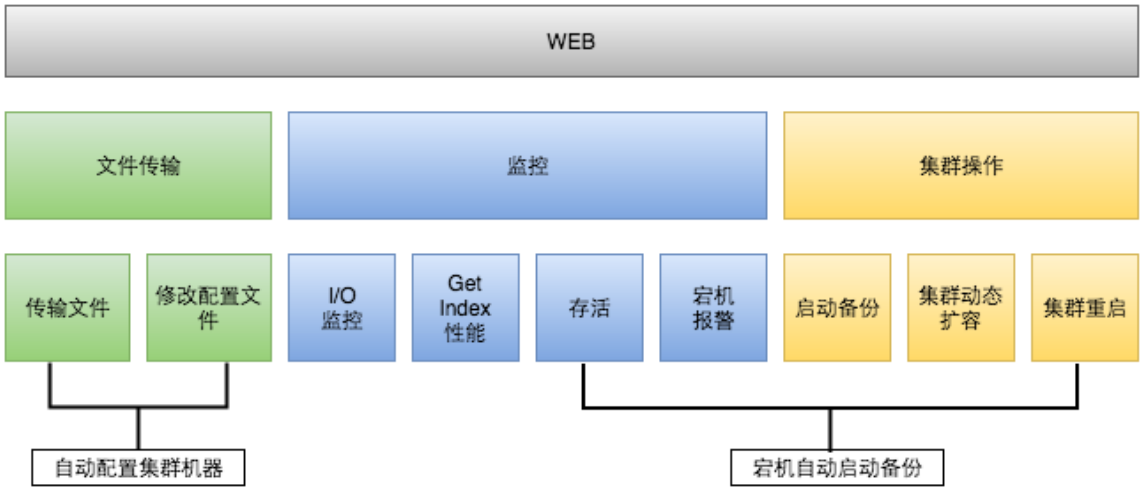
2)一般来讲，集群重启就可以恢复。这是需要增加ES的JVM堆内存。

3)最好及时发现，停止集群自动平衡分片功能，重启挂掉的机器，尽快加入集群。如果不能及时发现，可能导致恢复缓慢，如果同时还有写入，就可能出现数据不一致，很麻烦，同一个查询可能会出现不同的结果。

3.1 详细设计

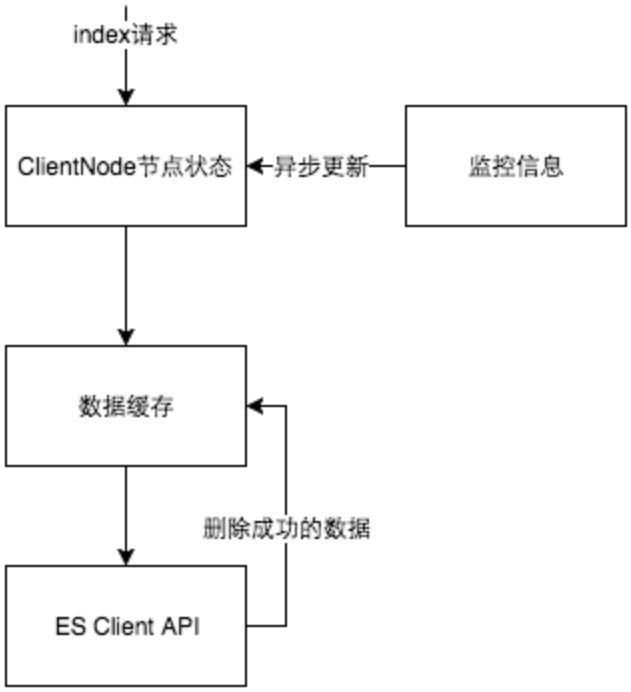
根据2.2的集群方案，我们需要完成以下几个功能

ES管理平台：



加强型ESClient：

增强Index


index请求
ClientNode节点状态
异步更新
监控信息
数据缓存
删除成功的数据
ES Client API