ALIBABA CLOUD

阿里云

阿里云Elasticsearch

常见问题

文档版本：20201124

阿里云

法律声明

阿里云提醒您在阅读或使用本文档之前仔细阅读、充分理解本法律声明各条款的内容。如果您阅读或使用本文档，您的阅读或使用行为将被视为对本声明全部内容的认可。

1. 您应当通过阿里云网站或阿里云提供的其他授权通道下载、获取本文档，且仅能用于自身的合法合规的业务活动。本文档的内容视为阿里云的保密信息，您应当严格遵守保密义务；未经阿里云事先书面同意，您不得向任何第三方披露本手册内容或提供给任何第三方使用。

2. 未经阿里云事先书面许可，任何单位、公司或个人不得擅自摘抄、翻译、复制本文档内容的部分或全部，不得以任何方式或途径进行传播和宣传。

3. 由于产品版本升级、调整或其他原因，本文档内容有可能变更。阿里云保留在没有任何通知或者提示下对本文档的内容进行修改的权利，并在阿里云授权通道中不时发布更新后的用户文档。您应当实时关注用户文档的版本变更并通过阿里云授权渠道下载、获取最新版的用户文档。

4. 本文档仅作为用户使用阿里云产品及服务的参考性指引，阿里云以产品及服务的“现状”、“有缺陷”和“当前功能”的状态提供本文档。阿里云在现有技术的基础上尽最大努力提供相应的介绍及操作指引，但阿里云在此明确声明对本文档内容的准确性、完整性、适用性、可靠性等不作任何明示或暗示的保证。任何单位、公司或个人因为下载、使用或信赖本文档而发生任何差错或经济损失的，阿里云不承担任何法律责任。在任何情况下，阿里云均不对任何间接性、后果性、惩戒性、偶然性、特殊性或刑罚性的损害，包括用户使用或信赖本文档而遭受的利润损失，承担责任（即使阿里云已被告知该等损失的可能性）。

5. 阿里云网站上所有内容，包括但不限于著作、产品、图片、档案、资讯、资料、网站架构、网站画面的安排、网页设计，均由阿里云和/或其关联公司依法拥有其知识产权，包括但不限于商标权、专利权、著作权、商业秘密等。非经阿里云和/或其关联公司书面同意，任何人不得擅自使用、修改、复制、公开传播、改变、散布、发行或公开发表阿里云网站、产品程序或内容。此外，未经阿里云事先书面同意，任何人不得为了任何营销、广告、促销或其他目的使用、公布或复制阿里云的名称（包括但不限于单独为或以组合形式包含“阿里云”、“Aliyun”、“万网”等阿里云和/或其关联公司品牌，上述品牌的附属标志及图案或任何类似公司名称、商号、商标、产品或服务名称、域名、图案标示、标志、标识或通过特定描述使第三方能够识别阿里云和/或其关联公司）。

6. 如若发现本文档存在任何错误，请与阿里云取得直接联系。

通用约定

格式	说明	样例
危险	该类警示信息将导致系统重大变更甚至故障，或者导致人身伤害等结果。	危险 重置操作将丢失用户配置数据。
警告	该类警示信息可能会导致系统重大变更甚至故障，或者导致人身伤害等结果。	警告 重启操作将导致业务中断，恢复业务时间约十分钟。
注意	用于警示信息、补充说明等，是用户必须了解的内容。	注意 权重设置为0，该服务器不会再接受新请求。
说明	用于补充说明、最佳实践、窍门等，不是用户必须了解的内容。	说明 您也可以通过按Ctrl+A选中全部文件。
>	多级菜单递进。	单击**设置**> **网络**> **设置网络类型**。
粗体	表示按键、菜单、页面名称等UI元素。	在**结果确认**页面，单击**确定**。
Courier字体	命令或代码。	执行 `cd /d C:/window` 命令，进入Windows系统文件夹。
斜体	表示参数、变量。	`bae log list --instanceid` *Instance_ID*
[] 或者 [a|b]	表示可选项，至多选择一个。	`ipconfig [-all|-t]`
{} 或者 {a|b}	表示必选项，至多选择一个。	`switch {active|stand}`

目录

1.集群负载不均问题的分析方法及解决方案

es负载不均

导致阿里云Elasticsearch（简称ES）的负载不均问题的原因很多，目前主要包括shard设置不合理、segment大小不均、冷热数据需求、负载均衡及多可用区架构部署的长连接不释放等。本文介绍ES集群负载不均问题的分析方法及解决方案。

问题现象

- 节点间磁盘使用率差距不大，监控中节点CPU使用率或load_1m呈现明显的负载不均衡现象。
- 节点间磁盘使用率差距很大，监控中节点CPU使用率或load_1m呈现明显的负载不均衡现象。

问题原因

- Shard设置不合理。

  > 注意　大多数负载不均问题是由于shard设置不合理导致，建议优先排查。

- Segment大小不均。
- 存在典型的冷热数据需求场景。

  > 注意　冷热数据场景（例如查询中添加了routing、查询频率较高的热点数据）必然导致数据出现负载不均。

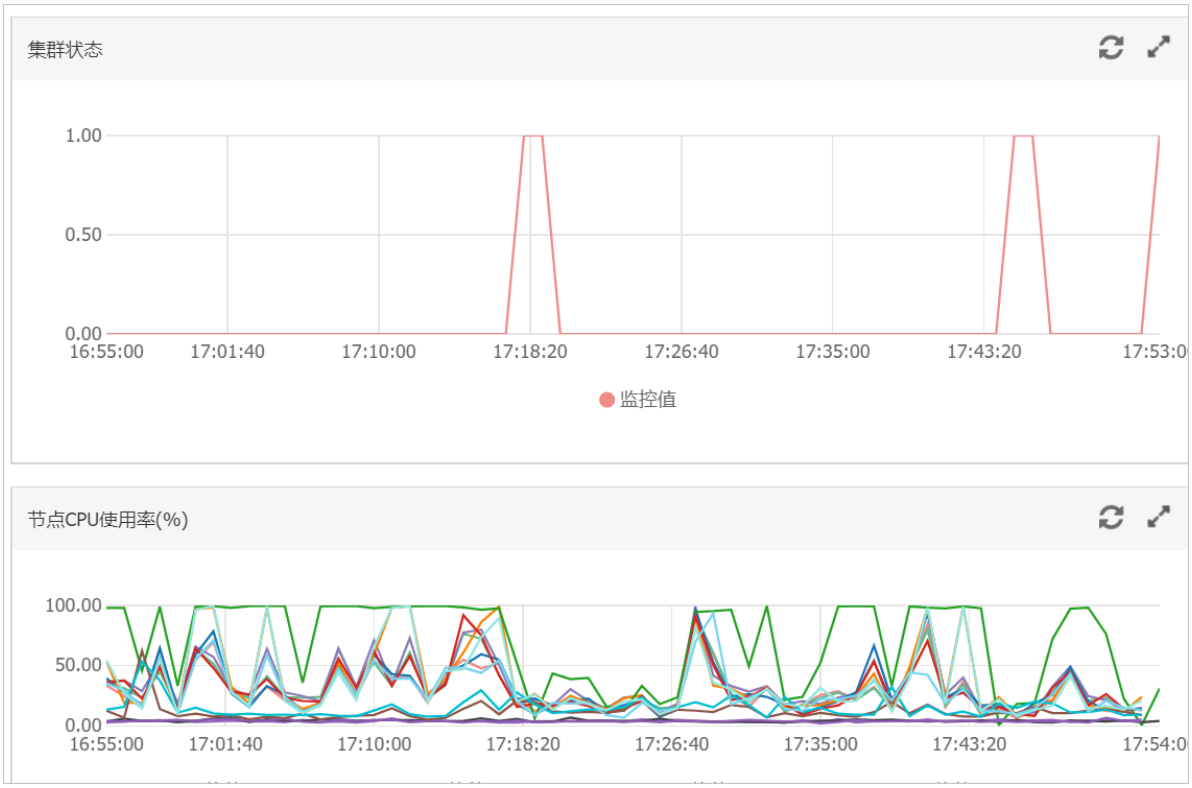
- 采用负载均衡及多可用区架构部署时，由于长连接不释放可能会导致流量不均（很少出现），详情请参见多可用区架构部署。

> 注意　其他情况导致的负载不均问题，请联系阿里云技术支持工程师排查。

Shard设置不合理

- 场景

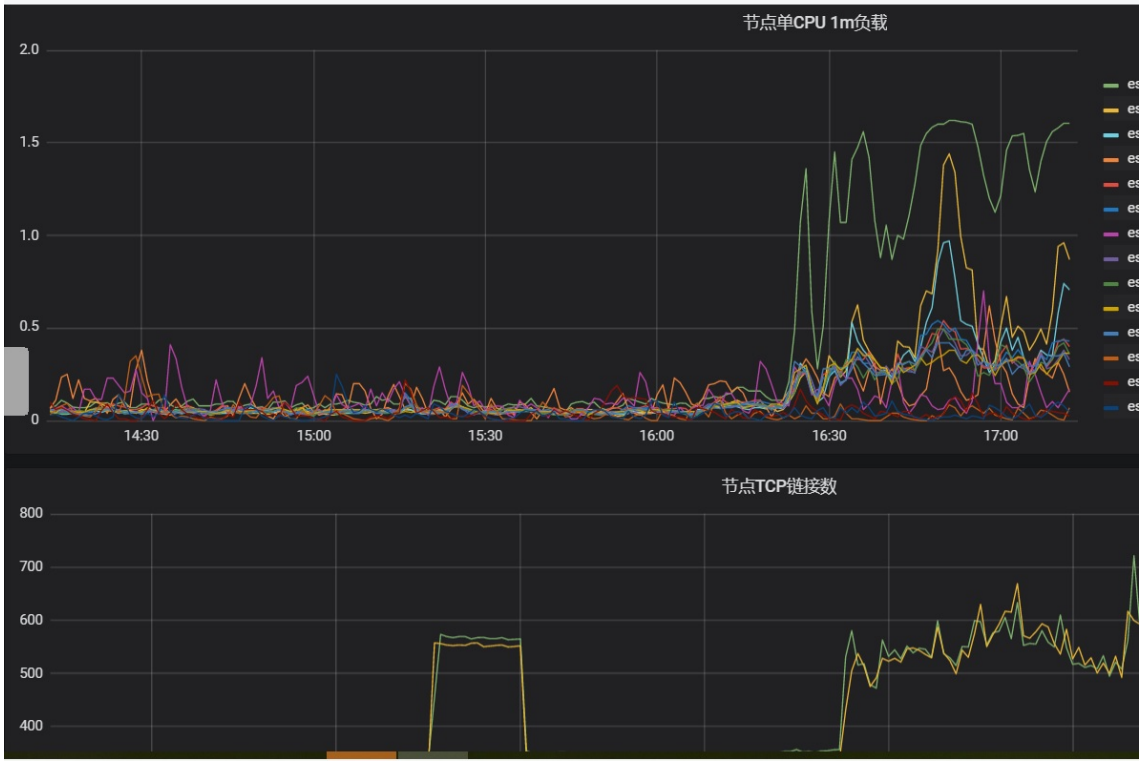
A公司有一个阿里云ES实例，该实例配置为：3个主节点16核32GB、9个数据节点32核64GB，主要数据保存在test索引上。当在业务高峰期的时候（16:21~18:00左右），查询QPS为2000左右（查询中没有冷热数据分离）、写入QPS为1000、2个节点的CPU达到100，负载过高影响ES服务。



- 分析

  i. 优先检查查询期间的网络及ECS情况。如果ECS环境正常，再查看网络流量监控。

  根据监控结果发现异常期间（16:21）网络请求上升，查询QPS上升，对应的CPU节点负载大幅增高。结合查询策略，初步判断负载高的节点主要承担了查询请求。



ii. 使用 `GET _cat/shards?v` ，查看索引的shard信息。

从结果可以看到test索引的shard在负载高的节点上呈现的数量较多，说明shard分配不均；然后查看磁盘使用率监控图，显示负载高的节点使用率比其他节点高。由此可以得出，shard分配不均衡导致存储不均，在业务查询或写入中，存储高的节点承担主要的查询和写入。

iii. 使用 `GET _cat/indices?v` ，查看索引信息。

从结果可以看到索引的主shard数量为5，副shard数量为1。结合集群配置，发现存在节点shard分配不均的现象，其次集群中存在被删除的文档。ES在检索过程中也会检索.del文件，然后过滤标记有.del的文档，大大地降低检索效率，耗费规格资源，建议在业务低峰期进行force merge。

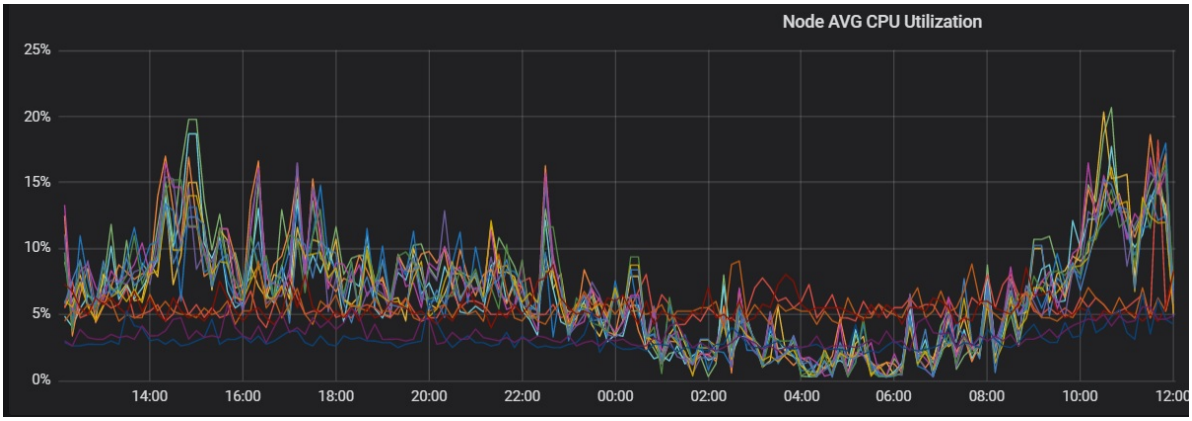
pri	rep	docs.count	docs.deleted	store.size	pri.store.size
1	1	8640	0	6.7mb	3.3mb
1	1	8639	0	6.7mb	3.3mb
1	1	8639	0	6.7mb	3.3mb
1	1	297847	488	688.7mb	344.8mb
1	1	93230	364	222.1mb	111mb
1	1	297803	588	693.5mb	346.1mb
1	1	297756	540	697.5mb	348.8mb
1	1	297852	536	687.9mb	343.4mb
1	1	297750	588	693.1mb	344.2mb
1	1	8639	0	6.7mb	3.3mb
1	1	8639	0	6.7mb	3.3mb
1	1	2	0	19.3kb	8.3kb
3	1	0	0	1.1kb	576b
3	1	71888299	15614684	124gb	64.6gb
3	1	0	0	1.1kb	576b
1	1	2661	0	2.2mb	1.1mb
1	1	8639	0	6.7mb	3.3mb
1	1	8639	0	6.6mb	3.3mb
1	1	297732	540	700.5mb	349.9mb
1	1	297795	540	691.6mb	344.7mb

iv. 查看主日志及searching慢日志。

从结果可以看到查询请求都是普通的term查询，且主日志正常，可以排除ES集群本身出现问题以及存在消耗CPU的查询语句的情况。

- 总结

通过以上分析，可以判断CPU负载不均主要是由于shard分布不均导致的。重新分配分片，确保主shard数与副shard数之和是集群数据节点的整数倍，集群的CPU负载趋于稳定。优化后的CPU趋势图如下。



- 解决方案

在创建索引时，合理规划shard，详情请参见Shard评估建议。

Shard评估建议

Shard大小和数量是影响ES集群稳定性和性能的重要因素之一。ES集群中任何一个索引都需要有一个合理的shard规划。合理的shard规划能够防止因业务不明确，导致分片庞大消耗ES本身性能的问题。

> **说明** ES 7.x以下版本的索引默认包含5个主shard，1个副shard；ES 7.x 及以上版本的索引默认包含1个主shard，1个副shard。

- 建议在小规格节点下，单个shard大小不要超过30GB。对于更高规格的节点，单个shard大小不要超过50GB。
- 对于日志分析或者超大索引场景，建议单个shard大小不要超过100GB。
- 建议shard的个数（包括副本）要尽可能等于数据节点数，或者是数据节点数的整数倍。

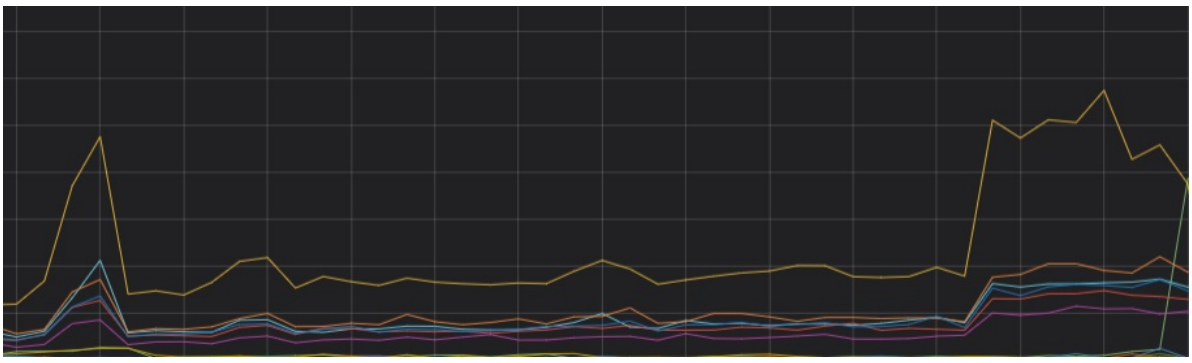
> **说明** 主分片不是越多越好，因为主分片越多，ES性能开销也会越大。

- 建议单节点shard总数按照单节点内存*30进行评估，如果shard数量太多，极易引起文件句柄耗尽，导致集群故障。
- 建议单个节点上同一索引的shard个数不要超5个。
- 如果您使用了自动创建索引功能，可通过设置场景模板，调整索引shard均衡，详情请参见修改场景化配置模板。

Segment大小不均

- 场景

  A公司ES集群忽然出现单个节点CPU飙升，影响查询性能。查询主要集中在test索引、shard设置为3主1副、shard分配均衡、索引中存在大量的delete.doc，且优先排除了底层主机问题。



- 分析

  i. 在查询body中添加 `"profile": true` 。

  根据结果可以看到test索引的1号shard查询时间比其他shard长。

  ii. 查询中分别指定 `preference=_primary` 和 `preference=_replica` ，body中添加 `"profile": true` ，分别查看主副shard查询消耗的时间。

  根据结果定位出索引的1号shard耗时主要体现在主shard上，判断和shard有关。

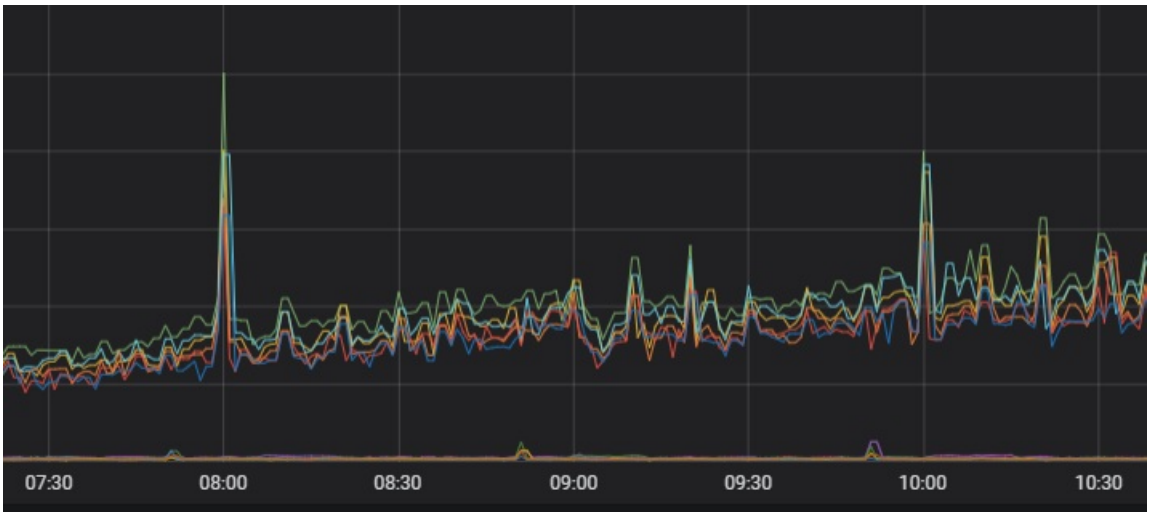
  iii. 使用 `GET _cat/segments/index?v&h=shard,segment,size,size.momery,ip` 及 `GET _cat/shards?v` ，查看索引的1号shard的信息。

  从结果可以看到索引的1号shard中存在比较大的segment，且doc数量比正常的副shard多。由此可以判断负载不均与segment大小不均有关。

> **说明** 造成主副shard的doc数量不一致的原因有很多，例如以下两种情况：
>
> - 如果一直有doc写入shard，由于主副数据同步有一定延迟，会导致数据不一致。但一般停止写入后，主副doc数量是一致的。
> - 如果使用自动生成docid的方式写入doc，由于主shard写入完成后会转发请求到副shard，因此在此期间，如果执行了删除操作，例如并行发送Delete by Query请求删除了主分片上刚写完的doc，那么副shard也会执行此删除请求；然后主shard又转发写入请求到副shard上，对于自动生成的id，doc将直接写入副shard，不进行检查，最终导致主副shard的doc数量不一致，同时在 `doc.delete` 中也可以看到主shard中存在大量的删除文档。

- 解决方案（任选一种）
  - 在业务低峰期进行force merge，将缓存中的delete.doc彻底删除，将小segment合并成大segment。
  - 重启主shard所在节点，触发副shard升级为主shard。并且重新生成副shard，副shard复制新的主shard中的数据，保持主副shard的segment一致。

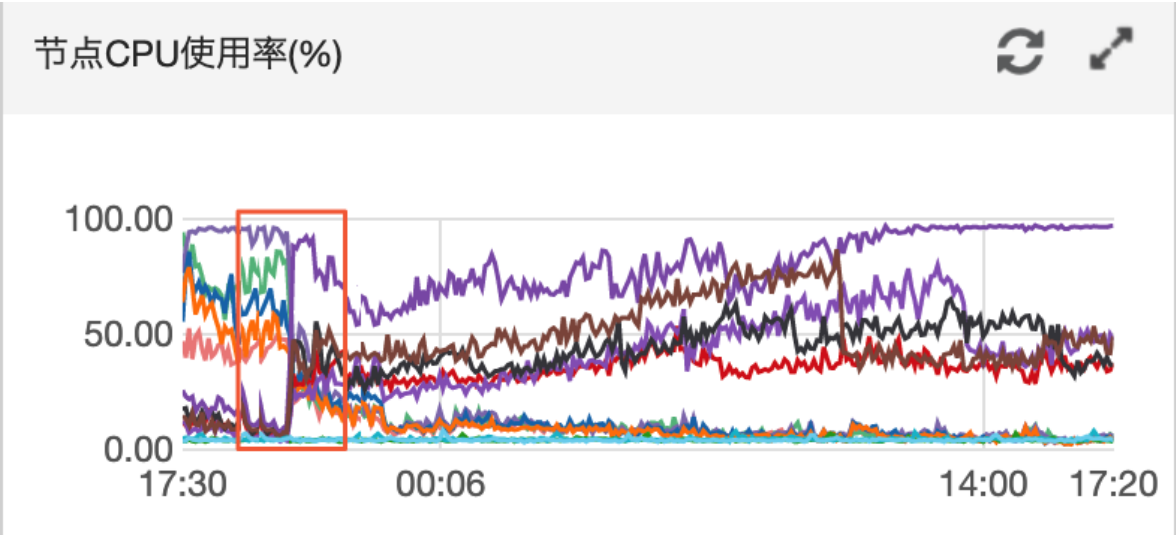
优化后的负载监控图如下。



多可用区架构部署

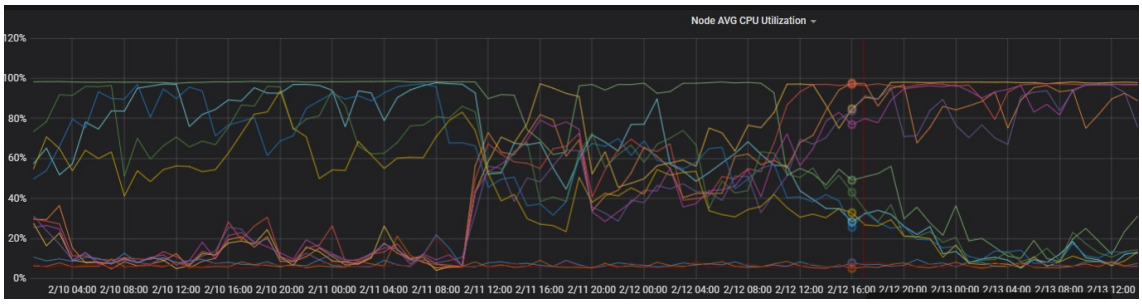
- 场景

A公司为实现ES集群跨区域容灾部署，分别在b区和c区部署高可用架构，在持续提供业务操作时，集群中c区节点负载明显比b区高，并且已经排除数据分配不均及硬件问题。



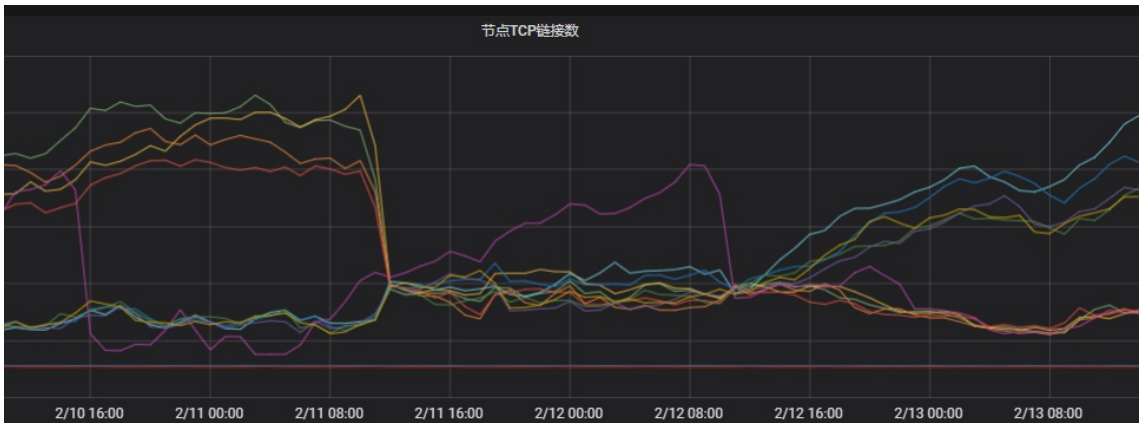
- 分析
  i. 观察近4天两区域节点的CPU监控。

  根据结果可以看到两区域节点CPU出现较大负载变动。



  ii. 观察节点TCP连接。

  根据结果可以看到两个区域下的连接数相差很大。判定与网络不均相关。

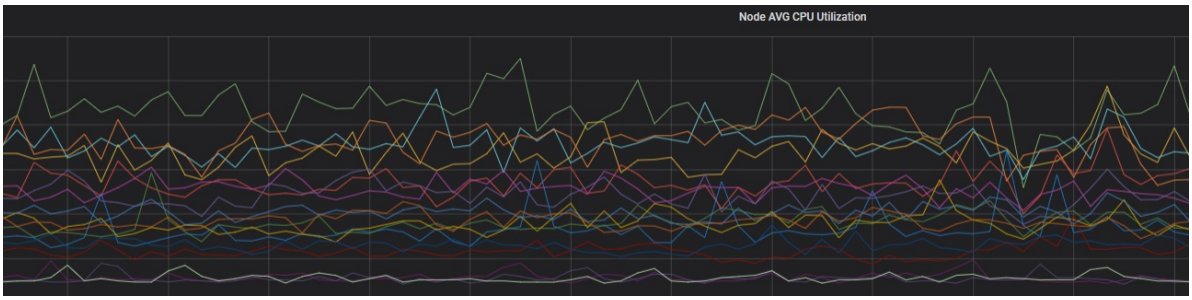


  iii. 排查客户端的链接情况。

客户端使用长连接，新建连接比较少，恰恰命中了多可用区网络独立调度的风险（网络服务是基于连接数独立调度的，每个调度单元会选择自己认为最优的节点进行创建，独立调度的性能会更高，但独立调度的风险是当新连接比较少的时候，有一定概率出现大部分调度单元都选择相同的节点去建立连接）。ES的协调节点对于请求的转发是同区域优先，即ES将请求转到其他区域的可能性较小，这样就导致当前区域出现负载不均的问题。

- 解决方案（任选一种）
  - 使用单独的协调节点，通过工作职责划分来降低风险。使用协调节点转发复杂流量，即使负载高，也不会影响到后面的数据节点。
  - 配置2套独立的域名，分别在客户端配置这2个域名，从而使客户端侧尽量保证区域的流量均等。该方案存在一定的风险，即高可用性会有一定的损失，后续进行集群升配时，由于节点不会从SLB移除，因此可能会出现访问失败的问题。

优化后的负载监控图如下。



2.集群磁盘使用率过高和read_only问题的排查与处理方法

本文介绍当您遇到磁盘使用率超过85%，甚至达到100%，导致阿里云Elasticsearch（简称ES）集群或Kibana无法正常提供服务时，采用的排查方式。

es集群磁盘使用率过高 es read_only问题 index read_only

> **注意** 免责声明：本文档可能包含第三方产品信息，该信息仅供参考。阿里云对第三方产品的性能、可靠性以及操作可能带来的潜在影响，不进行任何暗示或其他形式的承诺。

问题描述

- 在进行索引请求时，返回类似 `index read_only` 的报错。例如 `FORBIDDEN/12/index read-only / allow delete (api)];]` 。
- 集群处于Red状态，严重情况下存在节点未加入集群的情况（可通过 `GET _cat/nodes?` 查看），并且存在未分配的分片（可通过 `GET _cat/allocation?v` 查看）。

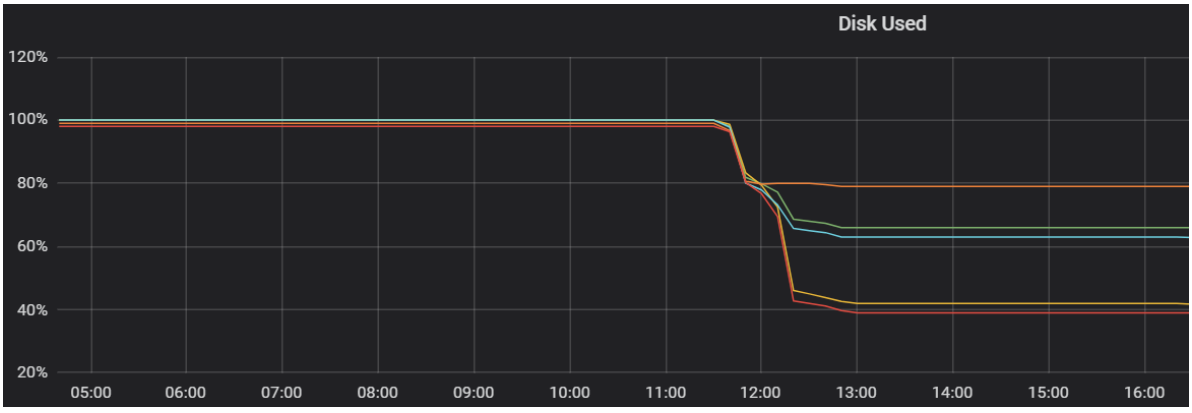
  > **说明** Red状态代表部分主分片不可用，可能已经丢失数据。

- 通过Kibana控制台创建管道、注册Beats时报错： `internal server error` 。
- 通过阿里云ES控制台的集群监控页面，或者登录Kibana控制台进入监控页面，动态查看近一段时间的集群负载，磁盘使用率曾达到100%。

问题原因

上述问题是由于磁盘使用率过高导致的。数据节点的磁盘使用率存在以下三个水位线，超过水位线可能会影响阿里云ES或Kibana服务：

- 超过85%，会导致新的分片无法分配。
- 超过90%，阿里云ES会尝试将对应节点中的分片迁移到其他磁盘使用率比较低的数据节点中。
- 超过95%，系统会对阿里云ES集群中的每个索引强制设置 `read_only_allow_delete` 属性，此时索引将无法写入数据，只能读取和删除对应索引。



解决方案

1. 执行以下命令删除数据。

   > 警告 数据删除后将无法恢复，请谨慎操作。您也可以选择保留数据，但需进行磁盘扩容，详情请参见升配集群。

   ```
   curl -u <username>:<password> -XDELETE http://<host>:<port>
   ```

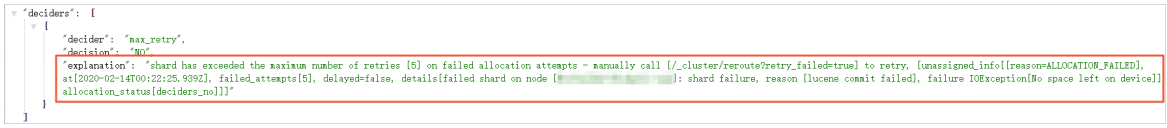
   - <host> 表示阿里云ES实例的公网或内网地址，建议使用前先确认相关白名单是否开启。
   - 执行命令后，如果集群无法响应，建议触发强制重启，在重启阶段尝试执行删除命令。

2. 检测集群索引是否依然为read_only状态，如果是，执行以下命令，将集群中所有索引的 index.blocks.read_only_allow_delete 属性设置为 null ，使集群中不再存在read_only状态的索引。

   ```
   PUT _settings
   {
     "index.blocks.read_only_allow_delete": null
   }
   ```

3. 检测集群是否依然为Red状态，如果是，可使用 _cat/allocation?v 命令查看集群中是否存在未分配的分片。

4. 如果存在未分配的分片，可执行 GET _cluster/allocation/explain 命令查看未分配分片的原因。如果原因如下图，请手动执行 POST /_cluster/reroute?retry_failed=true 命令。

5. 等待分片下发完成后，查看集群状态。如果集群状态依然为Red，请联系阿里云技术支持工程师解决。

更多信息

为避免磁盘使用率过高影响阿里云ES服务，建议开启磁盘使用率监控报警，及时查收报警短信，提前做好防御措施，详情请参见配置云监控报警。

3.Beats安装失败的排查与解决方法

当您安装采集器（Beats）时，遇到安装失败、心跳异常的情况时，可参考本文的方法进行排查解决。

操作步骤

1. 排查安装Beats服务的ECS的操作系统是否是Aliyun Linux、RedHat或CentOS。
2. 排查安装Beats服务的ECS是否与Elasticsearch或Logstash实例处于同一专有网络下。
3. 排查安装Beats服务的ECS是否安装了云助手和Docker。您可以连接ECS实例，通过以下命令进行验证。
   - 查看云助手服务状态

     ```
     systemctl status aliyun.service
     ```

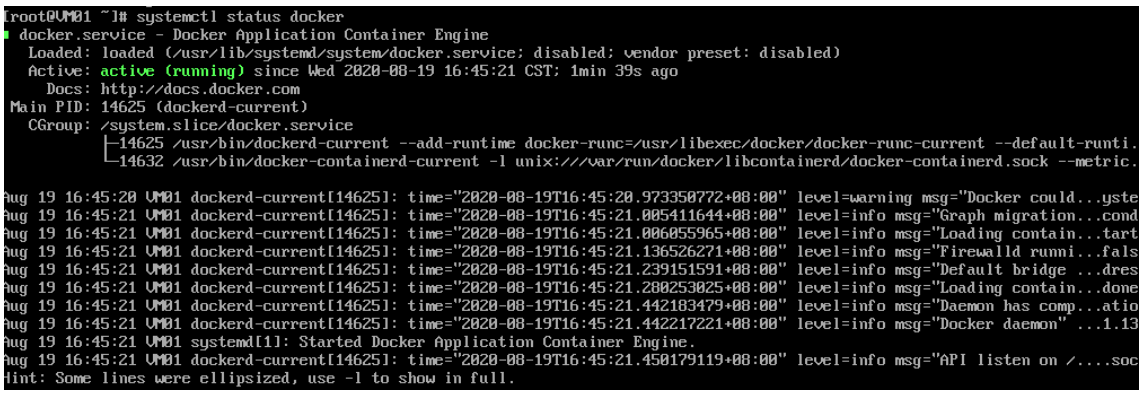
     正常情况下，返回结果如下。

     如果未安装云助手，可参见安装云助手客户端进行安装。
   - 查看Docker状态

     ```
     systemctl status docker
     ```

     正常情况下，返回如下结果。

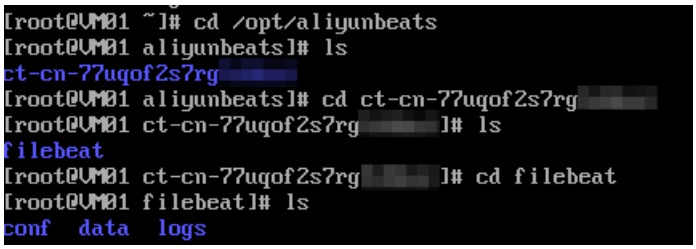
     如果未安装Docker，可参见部署并使用Docker（Alibaba Cloud Linux 2）进行安装。
4. 检查采集器的YML配置，确认是否已设置如下信息。

```
- type: log
 # Change to true to enable this input configuration.
   enabled: true
 # Paths that should be crawled and fetched. Glob based paths.
 paths:
  - /var/log/*.log
```

参数	说明
enabled	默认为false，使用时一定要设置为true。
paths	指定日志文件路径，可以采用模糊匹配，例如*.log。

> 注意
>
> - paths与配置页面填写的Filebeat文件目录存在区别。Filebeat文件目录是Docker映射的目录，只有映射到采集目录下才能采集到paths指定的文件。建议二者保持一致。
> - 在配置页面指定**采集器Output**后，YML配置下不能重新指定Output，否则会提示安装错误。
> - 对于采集器YML中默认已经注释掉（#）的参数，需谨慎修改，比如与X-Pack相关的参数设置，否则会导致安装失败。

5. 连接ECS实例，查看*/opt/aliyunbeats/*目录下，是否生成了对应的Beats实例，并确认是否存在conf、data、logs数据。



您也可以在logs目录下查看Beats日志，方便定位问题。

6. 查看Beats所在容器的运行状态，并分析日志定位问题。

   i. 查看Docker容器中的运行状态。

   ```
   docker ps -a | grep filebeat
   ```

ii. 当容器处于exited状态时，查看容器输出日志。

```
docker logs -f 容器id
```

4.通过手动迁移shard均匀分布热点数据的解决方案

Elasticsearch通过哈希映射将文档均匀地路由到分片中，同时shard均匀地分散在各个数据节点中，这样可能会出现某些节点存储的热点数据较多，导致这些节点的负载较高的情况。针对这种情况，可采用重启集群或手动迁移shard的方式，重新分配shard，临时降低高负载节点的压力。本文介绍如何手动迁移shard。

问题场景

- 负载高的节点中存在大量的同一属性分片，例如仅存在主分片。
- 业务索引在负载高的节点存储的shard比其他节点多。

注意事项

- 手动迁移shard，仅能临时解决节点压力较高的问题。如果节点短暂地脱离集群，shard重新分配，这些节点可能还会出现同样的问题。因此建议在迁移shard前，参见集群负载不均问题的分析方法及解决方案，优化shard后再迁移。
- 如果热点索引分配均衡，而集群整体压力较大，建议升配集群或扩容节点，解决资源紧张的问题。

解决方案

1. 禁用分片分配。

```
PUT /_cluster/settings
{
 "transient":{
   "cluster.routing.allocation.enable":"none"
 }
}
```

> 注意　以上命令仅临时禁用了分片分配，shard迁移完成后，需要重新启用（将cluster.routing.allocation.enable设置为all）。

2. 手动迁移shard。

   以下示例将index_parkingorder_v1索引，从192.168.130.77节点上的3号分片迁移到192.168.130.78节点上。

```
POST /_cluster/reroute
{
 "commands" : [
 {
  "move" : {
   "index" : "index_parkingorder_v1",
   "shard" : 3,
   "from_node" : "192.168.130.77",
   "to_node" : "192.168.130.78"
  }
 }
]
}
```

> **说明**
>
> - 迁移shard时，请确保同一序号的shard不能迁移到同一节点。例如，A节点中已经存在某一索引的3号副本分片，那么该索引的3号主分片就不能迁移到A节点中。
> - 手动迁移shard更详细的说明，请参见cluster-reroute。

3. 查看shard迁移状态。

```
GET _cat/shards?v
```

正常情况下，返回结果如下。

```
index      shard prirep state         docs   store ip         node
qosgeoname 4     r      STARTED    2279054 623.8mb 10.6.      y3m8a_2
qosgeoname 4     p      STARTED    2279054 628.3mb 10.6.      Xwhpjaf
qosgeoname 3     r      RELOCATING 2279002 627.3mb 10.6.      4Mh39az -> 10.6.      y3m8a_2tTdi_16TAy7B91w y3m8a_2
qosgeoname 3     p      STARTED    2279002 626.4mb 10.6.      Xwhpjaf
qosgeoname 1     r      STARTED    2278983 622.3mb 10.6.      I8SRFdj
qosgeoname 1     p      STARTED    2278983 621.6mb 10.6.      Qn3qSgY
qosgeoname 2     p      STARTED    2280694   623mb 10.6.      Qn3qSgY
qosgeoname 2     r      STARTED    2280694 622.7mb 10.6.      4Mh39az
qosgeoname 0     p      STARTED    2278770 624.7mb 10.6.      I8SRFdj
qosgeoname 0     r      STARTED    2278770 624.3mb 10.6.      y3m8a_2
```

4. shard迁移完成后，重新启用分片分配。

```
PUT /_cluster/settings
{
 "transient" : {
  "cluster.routing.allocation.enable" : "all"
 }
}
```