**数据统计文档**

V1.0.0

王智强 @ Enlink 数据中心

**目录**

[1. 概述 3](#_Toc22591)

[1.1. 说明 3](#_Toc17810)

[1.2. 统计环境 3](#_Toc902)

[2. res日志数据统计 4](#_Toc8589)

[2.1. 日志实例 4](#_Toc17084)

[2.2. 索引后数据的存储 4](#_Toc9363)

[2.3. 索引前后数据的对比 4](#_Toc18771)

[2.4. 日志数量 5](#_Toc27896)

[2.5. 资源占用率 5](#_Toc14929)

[3. translog 7](#_Toc8095)

[3.1. translog作用 7](#_Toc25883)

[3.2. translog生命周期 7](#_Toc23587)

[3.3. 利用translog进行数据恢复 7](#_Toc27010)

# 概述

## 说明

本文档描述日志原始数据以及被扫入Elasticsearch的数据统计和Filebeat扫描该日志时的资源占用率，资源占用率则为Filebeat运行时对系统CPU，IO，内存以及网络的占用情况。

## 统计环境

本文档统计所用到的日志源数据为CASBS后台EnWAS系统用户访问资源的操作记录（简称res日志），Elasticsearch集群由1个master主节点2个数据节点组成，配置为默认配置及5个分片1个副本分片。

本文中Filebeat的测试环境为VMware创建的虚拟服务器，虚拟服务器的操作系统为：CentOS Linux release 7.4.1708 (Core)。硬件配置信息为：CPU(英特尔 Xeon(至强) E5645 @ 2.40GHz 六核 (X2) )，内存（32G），硬盘（1.5T）。带宽1000Mbps（多台服务器公用）。

# res日志数据统计

## 日志实例

res日志为CASBS后台EnWAS系统用户访问资源的操作记录，每一条日志所占空间为350字节左右，具体日志实例如表2-1所示：

表 2-1 admin日志实例

|  |
| --- |
| 2018-12-04 13:40:49|WARNING|wangzq|zaf\_group|第三方用户|0254799d489846bc9a2674d8b8e03fac|192.168.32.181|www.baidu.com|1145|389|1534|DELETE|HTTP/1.1|www.baidu.com/images/index\_sy.js|404|128|text/xml|Mozilla/5.0 (Windows NT 10.0; Win64; x64) AppleWebKit/537.36 (KHTML, like Gecko) Chrome/67.0.3396.99 Safari/537.36|https://192.168.32.181:455/client/| |

## 索引后数据的存储

经观察Elasticsearch集群索引会存放在elasticsearch.yml配置文件指定的DATA路径下（通常为/var/lib/elasticsearch/nodes/0），该路径下的indices目录则存放Elasticsearch集群所存储的索引，每个索引存放在单独的文件夹中，文件夹名为索引名。具体结构如图2-1所示：

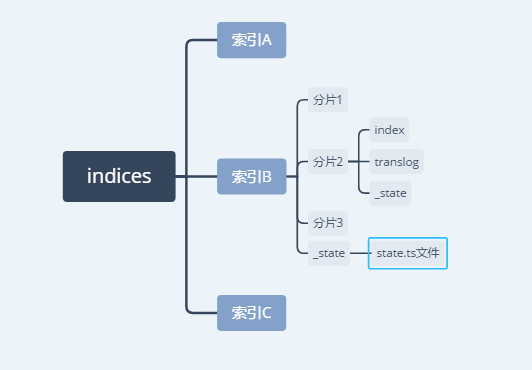


图 2-1 目录结构

index目录则为存储索引数据的地方，\_state目录存放state.ts文件，state.st则为当前分页的元信息（规则，位置，大小等信息），translog则存储临时文件以防止写入磁盘前系统崩溃从而造成缓存数据丢失。

## 索引前后数据的对比

因为当前Elasticsearch采用的是默认配置及5个分片1个副本分片，所以每个Elasticsearch节点都有可能包含0~4的分片数据，一部分为主分片另一部分为数据备份。数据前后对比如表2-2所示：

表 2-2 数据对比

|  |  |  |  |
| --- | --- | --- | --- |
| **所示日志所占空间（MB）** | **索引主分片所占空间(MB)** | **索引总分片所占空间（MB）** | **索引总分片所占空间/日志所占空间** |
| 11.95 | 6.7 | 13.5 | 1.13 |
| 18.92 | 10.1 | 20.4 | 1.08 |
| 91.87 | 43.3 | 86.9 | 0.95 |
| 197.81 | 94.3 | 188.3 | 0.95 |
| 440.63 | 207.4 | 414.2 | 0.94 |
| 846.50 | 395.5 | 790.5 | 0.93 |
| 1526 | 715.9 | 1331 | 0.87 |
| 2089 | 980.7 | 1946 | 0.93 |
| 3226 | 1434 | 2970 | 0.92 |
| 4547 | 2048 | 4198 | 0.92 |

Filebeat把admin日志推送到Elasticsearch集群后，日志数据在集群索引中所占空间为源日志所占空间的0.46倍，由于集群设置了副本分片也就是有数据备份，所以总分片所占空间是主分片的2倍及admin日志数据存储到Elasticsearch集群上所占空间为源日志大小的0.92倍。

## 日志数量

由于现阶段无法得知实际生产环境日志的产生速度，故模拟表2-3所述情况以供参考（每条日志所占空间按350字节每月按30天计算）：

表 2-3 日志数量及所占空间

|  |  |  |  |
| --- | --- | --- | --- |
| **日志产生速度（条/s）** | **每天/月产生日志数目（千万）** | **每天/月产生日志所占空间（G）** | **每天/月产生索引所占空间（G）** |
| 10 | 0.09/2.59 | 0.28/8.45 | 0.26/7.73 |
| 50 | 0.43/12.96 | 1.40/42.25 | 1.29/38.64 |
| 100 | 0.86/25.92 | 2.80/84.50 | 2.58/77.74 |
| 500 | 4.32/129.60 | 14.00/420.00 | 12.88/386.40 |
| 1000 | 8.64/259.20 | 28.00/840.00 | 25.76/772.80 |
| 2000 | 17.28/518.40 | 56.00/1680.00 | 51.52/1545.60 |
| 3000 | 25.92/777.60 | 84.00/2520.00 | 77.28/2318.40 |
| 5000 | 43.20/1296.00 | 140.00/4200.00 | 128.80/3864.00 |
| 8000 | 69.12/2073.60 | 224.00/6720.00 | 206.08/6182.40 |

## 资源占用率

表2-3则为Filebeat向Elasticsearch推送数据的不同速度下，Filebeat应用程序的进程在CPU，内存，IO，带宽使用率的占用情况。

表 2-4 资源占用率

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| **扫描速度（条/s）** | **CPU （6核）** | **内存 （32g）** | **磁盘读取 （kb/s）** | **磁盘写入 (kb/s)** | **发送请求**  **(kb/s)** | **接受请求 (kb/s)** |
| 0~150 | 1.5%~2.5% | 0.29% | 0 | 0~20 | 0~100 | 0~2 |
| 150~250 | 1.5%~2.5% | 0.30% | 0 | 0~20 | 100~150 | 0~5 |
| 250~500 | 2%~4% | 0.33% | 0 | 20~30 | 150~300 | 0~5 |
| 500~1000 | 4%~8% | 0.34% | 0 | 20~50 | 350~600 | 0~10 |
| 1000~1500 | 8%~15% | 0.34% | 0 | 20~50 | 500~800 | 10~15 |
| 1500~2000 | 10%~20% | 0.35% | 0 | 60~100 | 800~1200 | 10~20 |
| 2000~2500 | 15%~20% | 0.35% | 0 | 60~100 | 900~1800 | 15~30 |
| 2500~3500 | 15%~20% | 0.36% | 0 | 60~100 | 1000~2300 | 10~40 |
| 3500~4000 | 20%~45% | 0.36% | 0 | 60~150 | 1500~2500 | 20~50 |

参数注释：

CPU:进程对CPU的使用率，当前CPU为6核所以满负载时CPU使用率为600%

内存：进程使用的物理内存和总内存的百分比

磁盘读取：以kb为单位统计进程每秒从磁盘读的数量

磁盘写入:以kb为单位统计进程每秒写入磁盘的数量

发送请求：以kb为单位统计进程发送请求的流量

接受请求：以kb为单位统计进程接收请求的流量

# translog

## translog作用

为防止各种意外情况导致数据写入时候出现问题，Elasticsearch通过临时写入写操作来保证数据安全。因为集群索引过程中，数据会首先据缓存在内存中直到达到一个量（文档数或是占用空间大小）才会写入到磁盘。这就会带来一个风险，如果在写入磁盘前系统崩溃，那么这些缓存数据就会丢失。Elasticsearch通过translog解决了这个问题，每次写操作都会写入一个临时文件translog中，这样如果系统需要恢复数据可以从translog中读取。

所以在刚刚扫入一个索引时候，该索引所占空间要远远超出预测大小，其原因是每个分片下都会有translog数据缓存，其默认最大空间为512MB最长保留时间为12小时。

## translog生命周期

Elasticsearch 使用translog 来记录Elasticsearch中的操作，当新日志记录被添加到索引中时，新日志记录被加入到 in-memory buffer中，并且在translog中记录下来。当进行refresh操作时（默认情况下,es每隔一秒钟执行一次refresh，可以通过参数index.refresh\_interval来修改这个刷新间隔），in-memory buffer里的数据被清空，translog保持不变。经过一段时间，或者translog大到一定程度，整个index会被flushed，这时translog将被删除。

## 利用translog进行数据恢复

如果Elasticsearch需要恢复数据但已无可以成功恢复数据的副本分片，可以通过该分片所对应的translog来恢复其tranlog所包含的数据，但translog中的所有数据将会丢失。所用命令为：

|  |
| --- |
| bin/elasticsearch-translog truncate -d /var/lib/elasticsearchdata/nodes/0/indices/<index name>/<shard>/translog/ |