elasticsearch esrally 性能测试实操

最新在用 esrally 测试 es 的性能，今天把相关操作记录下。本人非专业测试，各位大佬请轻喷。

关于 esrally 的文档，请移步：[esrally测试](http://cxyroad.com/”https://esrally.readthedocs.io/en/latest/”)

esrally 是个 elastic 官方的测试工具，可以对 es 进行压力测试。其运行对环境有一定要求，如 python版本，JDK要求等，这里为了不去配环境，采用 docker 跑 esrally，所以首先你需要拉取 esrally 的官方镜像：

```
docker pull elastic/esrally
```

准备数据

接下来就是准备测试数据。

这里为了方便，也是采用常用的测试数据，请下载这里的数据：[documents-2.json](http://cxyroad.com/”https://link.zhihu.com/?target=http%3A//benchmarks.elasticsearch.org.s3.amazonaws.com/corpora/geonames/documents-2.json.bz2”)

下载后是个 bz2 的压缩文件，请先解压。

因为这块的测试数据时 geonames，实际字段不多，就是简单的地理位置信息和人口信息等。

主要最后解压解包出来应该是：`documents-2.json`。

接下来定义 index.json，这里主要是对测试数据的字段定义：
`index.json`

```
{
  "settings": {
    "index.number_of_replicas": 0
  },
  "mappings": {
    "docs": {
      "dynamic": "strict",
      "properties": {
        "geonameid": {
          "type": "long"
        },
        "name": {
          "type": "text"
        },
        "latitude": {
          "type": "double"
        },
        "longitude": {
          "type": "double"
        },
        "country_code": {
          "type": "text"
        },
        "population": {
          "type": "long"
        }
      }
    }
  }
}
```

按照 esrally 的测试要求，定义 track 信息：
`track.json`

```
{
  "version": 2,
  "description": "Tutorial benchmark for Rally",
```

```
  "indices": [
    {
      "name": "geonames",
      "body": "index.json",
      "types": [ "docs" ]
    }
  ],
  "corpora": [
    {
      "name": "rally-tutorial",
      "documents": [
        {
          "source-file": "documents-2.json", # 测试文件名
          "document-count": 11396503, # 测试数据大小，wc -l
documents-2.json
          "uncompressed-bytes": 3547613828 # 测试数据的字节数
        }
      ]
    }
  ],
  "schedule": [ # 以下定义了不同的测试，如对索引的创建与删除，bulk插入
，query-match，GC情况等
    {
      "operation": {
        "operation-type": "delete-index"
      }
    },
    {
      "operation": {
        "operation-type": "create-index"
      }
    },
    {
      "operation": {
        "operation-type": "cluster-health",
        "request-params": {
          "wait_for_status": "green"
        },
        "retry-until-success": true
      }
    },
    {
      "operation": {
        "operation-type": "bulk",
        "bulk-size": 5000
      },
      "warmup-time-period": 120,
      "clients": 8
```

```
    },
    {
      "operation": {
        "operation-type": "force-merge"
      }
    },
    {
      "operation": {
        "name": "query-match-all",
        "operation-type": "search",
        "body": {
          "query": {
            "match_all": {}
          }
        }
      },
      "clients": 8,
      "warmup-iterations": 1000,
      "iterations": 1000,
      "target-throughput": 100
    }
  ]
}
```

以上都是数据准备，接下来就可以通过 docker run 直接跑 esrally 测试了。

docker 测试环境准备

我的是参照其他人的：

```
# ls -R myrally/
myrally/: # 主要
benchmarks  logging.json  logs  rally.ini

myrally/benchmarks:
data  races  result.csv  result.singleton.csv  tracks

myrally/benchmarks/data:
single
```

myrally/benchmarks/data/single:
documents-2.json index.json rally-track-data-geonames.tar track.json

myrally/benchmarks/races: # 测试结果存放
16019f05-76e7-467b-ae3c-ac0e409d28a0 5cd3f772-3fa7-4b1a-bae6-d44451112f80 c45faf69-1c48-409d-be63-087f1b3d982d
d80fbbcf-f170-4c12-9d47-654ea3dc896b
356da763-4624-4e15-9109-79b77a7ef330 7baec91f-7e4b-47e7-88d0-624d76a95de3 d2b37a0d-bd29-4ac0-9417-dbcb1e61246f
e26d590e-d05c-4e19-b93d-40a7d53cbb41

myrally/benchmarks/races/16019f05-76e7-467b-ae3c-ac0e409d28a0:
race.json

myrally/benchmarks/races/356da763-4624-4e15-9109-79b77a7ef330:
race.json

myrally/benchmarks/races/5cd3f772-3fa7-4b1a-bae6-d44451112f80:
race.json

myrally/benchmarks/races/7baec91f-7e4b-47e7-88d0-624d76a95de3:
race.json

myrally/benchmarks/races/c45faf69-1c48-409d-be63-087f1b3d982d:
race.json

myrally/benchmarks/races/d2b37a0d-bd29-4ac0-9417-dbcb1e61246f:
race.json

myrally/benchmarks/races/d80fbbcf-f170-4c12-9d47-654ea3dc896b:
race.json

myrally/benchmarks/races/e26d590e-d05c-4e19-b93d-40a7d53cbb41:
race.json

myrally/benchmarks/tracks: # 主要
default tracktest

myrally/benchmarks/tracks/default:

myrally/benchmarks/tracks/tracktest: # 相关测试数据与定义，主要就是documents、index.json、track.json
documents-2.json documents-2.json.offset index.json rally-track-data-geonames.tar track.json track.json.bak

myrally/logs:
rally.log

```

正式测试
====

准备好后，测试下：

```
docker run --rm -v /home/test/rally/:/rally/.rally/ elastic/rally list tracks --track-path=/rally/.rally/benchmarks/data/single
```

然后是正式跑测试数据：
我的环境是 es集群在一个环境，测试放另一个环境。

docker run 命令有点长，放在 shell 脚本中：

```
#!/bin/bash

docker run --privileged=true  --rm   -v /root/myrally:/rally/.rally elastic/rally race --pipeline=benchmark-only --target-hosts=172.22.xxx.xxx:9200 --track-path=/rally/.rally/benchmarks/tracks/tracktest  --client-options="basic_auth_user:'xxx',basic_auth_password:'xxx'" --offline  --report-format=csv --report-file=/rally/.rally/benchmarks/result.csv
```

主要主要这几个配置：

```
--pipeline=benchmark-only 仅性能测试
--target-hosts=172.22.xxx.xxx:9200 测试的es集群
--track-path=/rally/.rally/benchmarks/tracks/tracktest 定义的测试项
--client-options="basic_auth_user:'xxx',basic_auth_password:'xxx'" 如果你定义了es的xpack认证的话
--offline 离线
--report-format=csv 输出格式
```
```

--report-file=/rally/.rally/benchmarks/result.csv 输出到具体文件

```

然后直接 运行脚本 就可以看到：

```
sh docker_run.sh

    ____   ____
   / __ ____ _/ / /_ __
  / /_/ / __ `/ / / / / /
 / _, _/ /_/ / / / /_/ /
/_/ |_|__,_/_/_/__, /
                /____/

[WARNING] merges_total_time is 4855 ms indicating that the cluster is
not in a defined clean state. Recorded index time metrics may be
misleading.
[WARNING] indexing_total_time is 8167 ms indicating that the cluster is
not in a defined clean state. Recorded index time metrics may be
misleading.
[WARNING] refresh_total_time is 20774 ms indicating that the cluster is
not in a defined clean state. Recorded index time metrics may be
misleading.
Running delete-index   [100%
done]
Running create-index   [100%
done]
Running cluster-health   [100%
done]
Running force-merge   [100%
done]
Running query-match-all   [100%
done]
Running agg   [100% done]
[INFO] Racing on track [tracktest] and car ['external'] with version
[6.8.23].

    _______   __   _____
   / ____(_)___ ____ _/ / / ___/_________ ________
  / /_ / / __ \/ __ `/ /   __ \/ ___/ __ \/ ___/ _ \
 / __/ / / / / / /_/ / /   ___/ / /__/ /_/ / / / __/
/_/   /_/_/ /_/__,_/_/   /____/___/____/_/   ___/
```
```

Metric,Task,Value,Unit
Cumulative indexing time of primary shards,,0.1457000000000002,min
Min cumulative indexing time across primary shards,,0,min
Median cumulative indexing time across primary shards,,0,min
Max cumulative indexing time across primary shards,,0.1212833333333334,min
Cumulative indexing throttle time of primary shards,,0,min
Min cumulative indexing throttle time across primary shards,,0,min
Median cumulative indexing throttle time across primary shards,,0,min
Max cumulative indexing throttle time across primary shards,,0,min
Cumulative merge time of primary shards,,0.08978333333333333,min
Cumulative merge count of primary shards,,70,
Min cumulative merge time across primary shards,,0,min
Median cumulative merge time across primary shards,,0,min
Max cumulative merge time across primary shards,,0.07588333333333333,min
Cumulative merge throttle time of primary shards,,0,min
Min cumulative merge throttle time across primary shards,,0,min
Median cumulative merge throttle time across primary shards,,0,min
Max cumulative merge throttle time across primary shards,,0,min
Cumulative refresh time of primary shards,,0.37195,min
Cumulative refresh count of primary shards,,603,
Min cumulative refresh time across primary shards,,0,min
Median cumulative refresh time across primary shards,,0,min
Max cumulative refresh time across primary shards,,0.2579666666666667,min
Cumulative flush time of primary shards,,0.006866666666666666,min
Cumulative flush count of primary shards,,2,
Min cumulative flush time across primary shards,,0,min
Median cumulative flush time across primary shards,,0,min
Max cumulative flush time across primary shards,,0.003983333333333335,min
Total Young Gen GC time,,0,s
Total Young Gen GC count,,0,
Total Old Gen GC time,,0,s
Total Old Gen GC count,,0,
Store size,,0.0034995889291167 26,GB
Translog size,,0.011255173012614 25,GB
Heap used for segments,,0.19719028472900 39,MB
Heap used for doc values,,0.14591598510742188,MB
Heap used for terms,,0.04439640045166 0156,MB
Heap used for norms,,0.0003662109375,MB
Heap used for points,,0.00133132934570 3125,MB
Heap used for stored fields,,0.00518035888671875,MB
Segment count,,17,
Total Ingest Pipeline count,,0,

```
Total Ingest Pipeline time,,0,s
Total Ingest Pipeline failed,,0,
Min Throughput,query-match-all,99.86,ops/s
Mean Throughput,query-match-all,99.94,ops/s
Median Throughput,query-match-all,99.94,ops/s
Max Throughput,query-match-all,99.95,ops/s
50th percentile latency,query-match-all,42.15359699446708,ms
90th percentile latency,query-match-all,47.279874711239245,ms
99th percentile latency,query-match-all,103.72783147904556,ms
99.9th percentile latency,query-match-all,242.50521747623142,ms
100th percentile latency,query-match-all,375.2201080060331,ms
50th percentile service time,query-match-all,39.29002500080969,ms
90th percentile service time,query-match-all,43.98797630856279,ms
99th percentile service time,query-match-all,53.264984017150724,ms
99.9th percentile service time,query-match-all,163.50637127645325,ms
100th percentile service time,query-match-all,372.0516809989931,ms
error rate,query-match-all,0.00,%
Min Throughput,agg,10.92,pages/s
Mean Throughput,agg,10.92,pages/s
Median Throughput,agg,10.92,pages/s
Max Throughput,agg,10.92,pages/s
100th percentile latency,agg,90.67862101073842,ms
100th percentile service time,agg,90.67862101073842,ms
error rate,agg,0.00,%

[INFO] Race id is [d2b37a0d-bd29-4ac0-9417-dbcb1e61246f]

--------------------------------
[INFO] SUCCESS (took 212 seconds)
--------------------------------
```

后面可以对 csv 文件处理下：

```
import pandas as pd

def csv2excel_cluster():
    csv = pd.read_csv("result.csv", encoding='utf-8')
    csv.to_excel("result.xlsx", sheet_name="cluster")
```

参考：

* 1.[ESRally离线安装与测试](http://cxyroad.com/”https://zhuanlan.zhihu.com/p/586411188”)
* 2.[esrally文档](http://cxyroad.com/”https://esrally.readthedocs.io/en/stable/migrate.html”)
* 3.[esrally 如何进行简单的自定义性能测试？](http://cxyroad.com/”https://mp.weixin.qq.com/s?__biz=MzI2NDY1MTA3OQ==&mid=2247489071&idx=1&sn=9306b9465f5ad742b9afda4ab7d18b55&chksm=eaa83e07dddfb711a224faafe6f4b11b223915c3e02ad47ac8a6d3dbc1d5859a807d7e3052fd&scene=21#wechat_redirect”)

原文链接: https://juejin.cn/post/7350717832449835019