# 1. Elasticsearch 性能基准测试

## 1.1. 官方推荐机器配置

- 内存
  es 最重要的资源是内存，64GB RAM 的机器是官方推荐的，32、16GB 的机器也能用于生产环境，但不建议小于 8GB。
- CPU
  es 对 CPU 开销不是很大 (本机开了 3 个 node，同时进行 100 并发查询、数据插入索引，才把 cpu 将将打满 100%)，但多核 CPU 还是必须的，通用的集群使用 2 到 8 核机器。
- 硬盘
  任何集群的瓶颈，对于磁盘 io 要求很高（如：大量、实时写入，大量存取索引），最好能使用 SSD。
- 网络
  由于 elasticsearch 的特性，集群节点间的传输，包括数据的分片、备份、同步，以及数据从不同节点的分片获取和汇总。网络带宽也是很重要的。

## 1.2. 性能试验方案

### 1.2.1. 数据库导入

采用 logstash 导入，以及 kettle 导入，并比较两者效率。主要测试磁盘 io、网络。

### 1.2.2. 本地 json 导入

通过本地 json 文件导入。主要测试 index 速度和效率。

### 1.2.3. 内存

1、利用 jstat 日志来得到内存指标的变化，`jstat -gc -h5 XXX 3s > test-100W.log`，不同数据量修改文件名，输出到不同文件中。
2、粗粒度的监控 Java Heap、GC，采用 Grafana 监控模式查看，通过 http resuful api 调用（查询）或 siege 压测的方式查看操作时的内存指标变化。
3、也可通过 Visualvm 的 Heap dump 查看相关指标。

#### 1.2.4. 查询效率

通过 siege、multiprocess 检测查询平均耗时。以及测试优化点。

#### 1.2.5. 官方的 benchmark 指标检测

配置 esrally 环境。https://github.com/elastic/rally

### 1.3. 数据准备

已准备 mysql 的数据表，数据规模 130 万。可扩展。已准备与上面相同的 json 文件，数据规模 130 万。可扩展。

### 1.4. 相关脚本

例如下面，为 es 创建 index 脚本，其他脚本见文件夹下：

```
curl -XPUT "http://127.0.0.1:9200/productindex"
curl -XPOST "http://127.0.0.1:9200/productindex/product/_mapping?pretty" -d '
{
    "product": {
            "properties": {
                "company_name": {
                    "type": "text",
                              "analyzer": "ik_smart",
                    "search_analyzer": "ik_smart"
                },
                "id": {
                    "type": "long"
                },
                "data_status": {
                    "type": "long"
                }
            }
        }
  }'
```

create*index**.sh：创建 index 以及 index 配置。
*logstash.conf: logstash 配置从 mysql 同步至 elasticsearch
multi_search.py: 多进程压测
elasticsearch_metrics.py: 循环获取系统指标供 grafana 监控。

Enjoy! By Hanbing.
Using madoko.