

轨迹相似查询性能测试

测试数据

从 12306 爬取 8000 多条真实火车轨迹。对于每一条火车轨迹，按照如下规则随机生成新的轨迹:

- 对于每个站台，选择以该站台为中心，10km 范围内的任意一点代表该站台。
- 每个站台都采用随机后的点表示。

具体的测试集大小参考各个测试。

mapping 如下:

日期 (date)	车次 (name)	类型 (type)	始发站 (from_station)	终点站 (to_station)	路线(path)	发车时间 (start_time)	站台 (stations)
date	string	string	string	string	geo_shape	long	string

测试数据示例:

日期 (date)	车次 (name)	类型 (type)	始发站 (from_station)	终点站 (to_station)	路线(path)	发车时间 (start_time)	站台 (stations)
2019-02-02	T297	T	北京	牡丹江	LineString(...)	[...,...]	["北京", "唐山北", ..., "海林", "牡丹江"]

集群配置

- 机器配置: 32 core, 3 * 1T sata, 128g 内存
- data 节点配置: jvm 堆大小为 32g
- index 配置:
 - `train`: 6 shard, 0 replica, refresh_interval 为 1min
 - `train_single`: 1 shard, 0 replica, refresh_interval 为 1min

以下所有测试均在单台 data 节点中运行。

示例查询

```
curl -XPOST -s "localhost:9201/${index}/_search?pretty&size=${size}" -d'
{
  "query": {
    "trajectory_simple_nearest_neighbour_query": {
      "size": 1000,
      "path_field": "path",
      "time_field": "start_time",
      "similarity_method": "lcss",
```

```
    "query": {
      "coordinates": [...],
      "datetimes": [...]
    }
  }
}
```

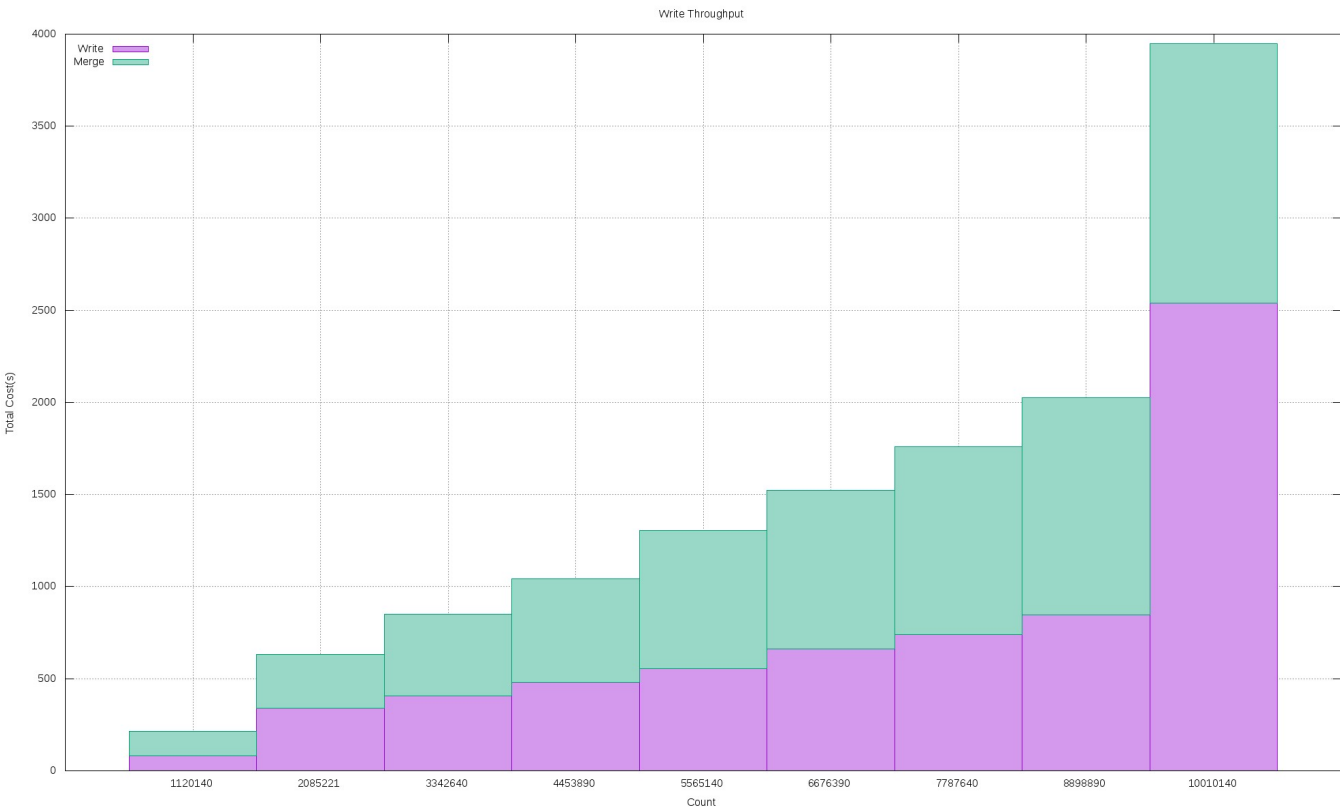
查询的轨迹从 index 中随机选出 100 个; 每一轮会依次以随机选出的轨迹作为输入执行查询, 执行 10 轮后计算平均响应时间。

Shard 性能

使用 `train_single` 这个 index 测试单个 shard 的性能. 通过构造不同大小的数据集, 测试查询/写入性能和索引常驻内存的使用大小。

写入性能

客户端以 bulk 方式写入, 每 1000 个请求构成 1 个 bulk, 并发为 4.



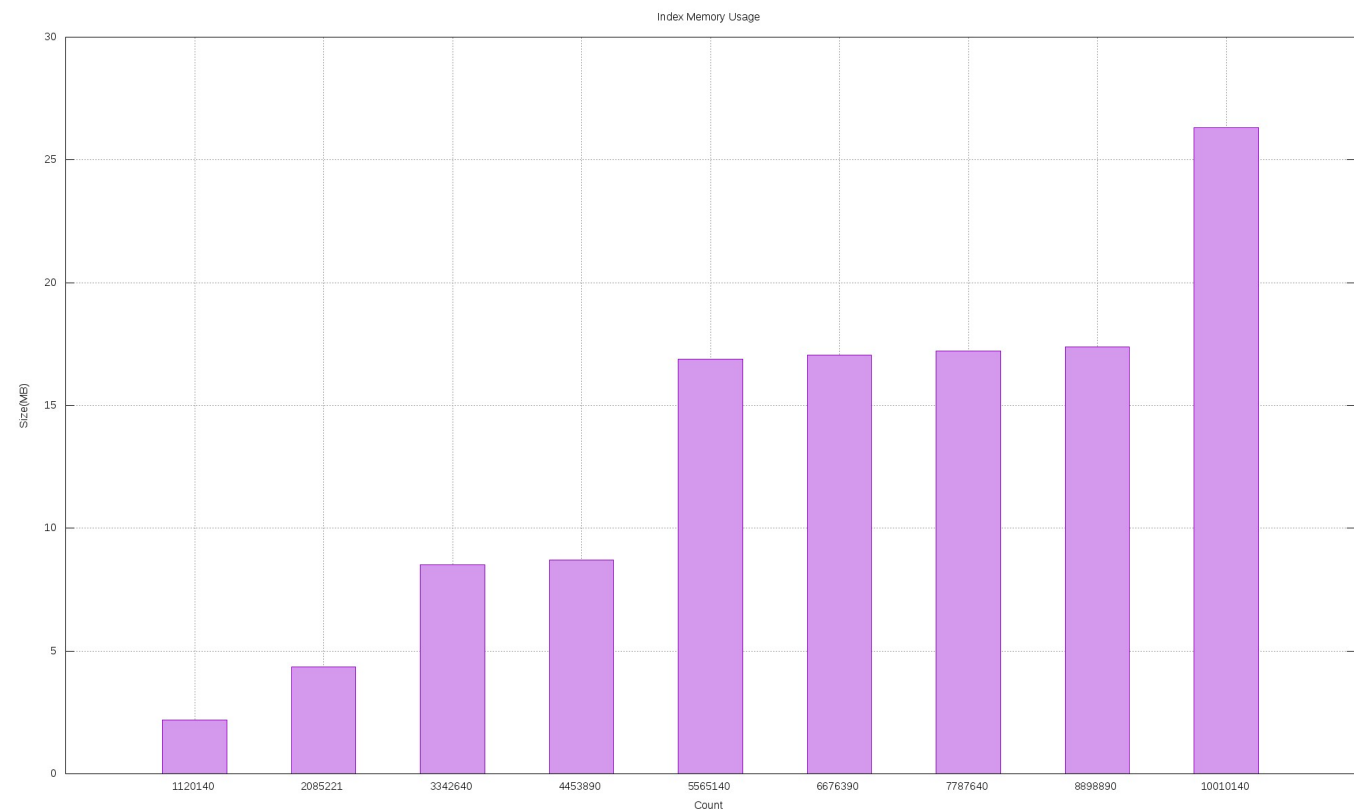
横轴表示数据集大小, 纵轴表示写入耗时(单位为秒)。上图统计了两个指标, 其中:

- **Write:** 表示客户端写入完成的耗时
- **Merge:** 表示 merge 为 1 个 segment 的耗时

单个 shard 写入过程中, 磁盘使用率在 30%以下, cpu 使用率在 40%以下。

内存使用

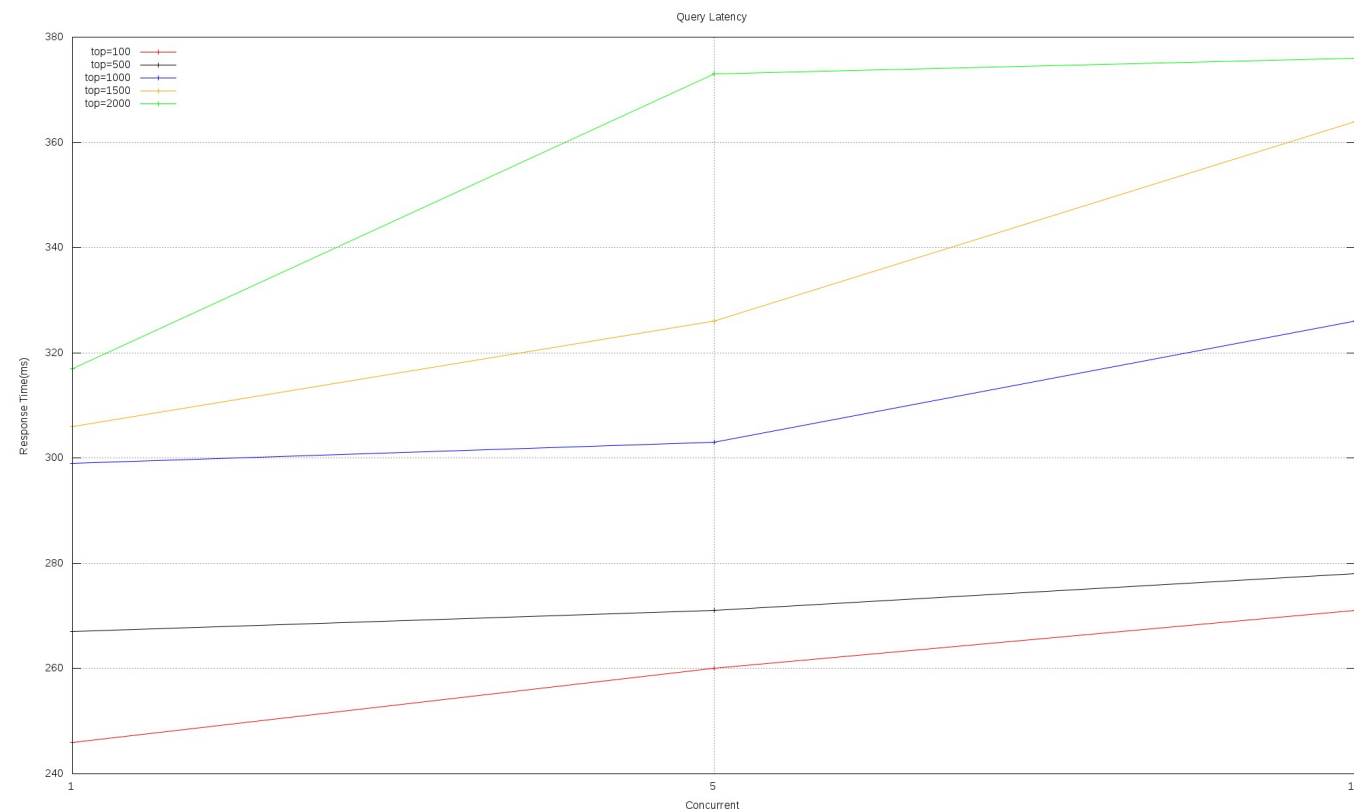
内存占用统计如下图:



横轴表示数据集大小，纵轴表示轨迹相关索引常驻内存大小(单位为 mb)。

查询性能

在 1kw 的测试集下，测试不同并发的响应时间



横轴表示客户端的并发数，纵轴表示查询响应时间(单位为 ms)。不同的折线表示 size 取不同值的结果。

在 10 个并发的情况下执行对train_single的查询(1 个 shard), cpu 的利用率在 30-40%.

Data 节点性能

使用 **train** 这个 index 测试单个 data 节点的性能。

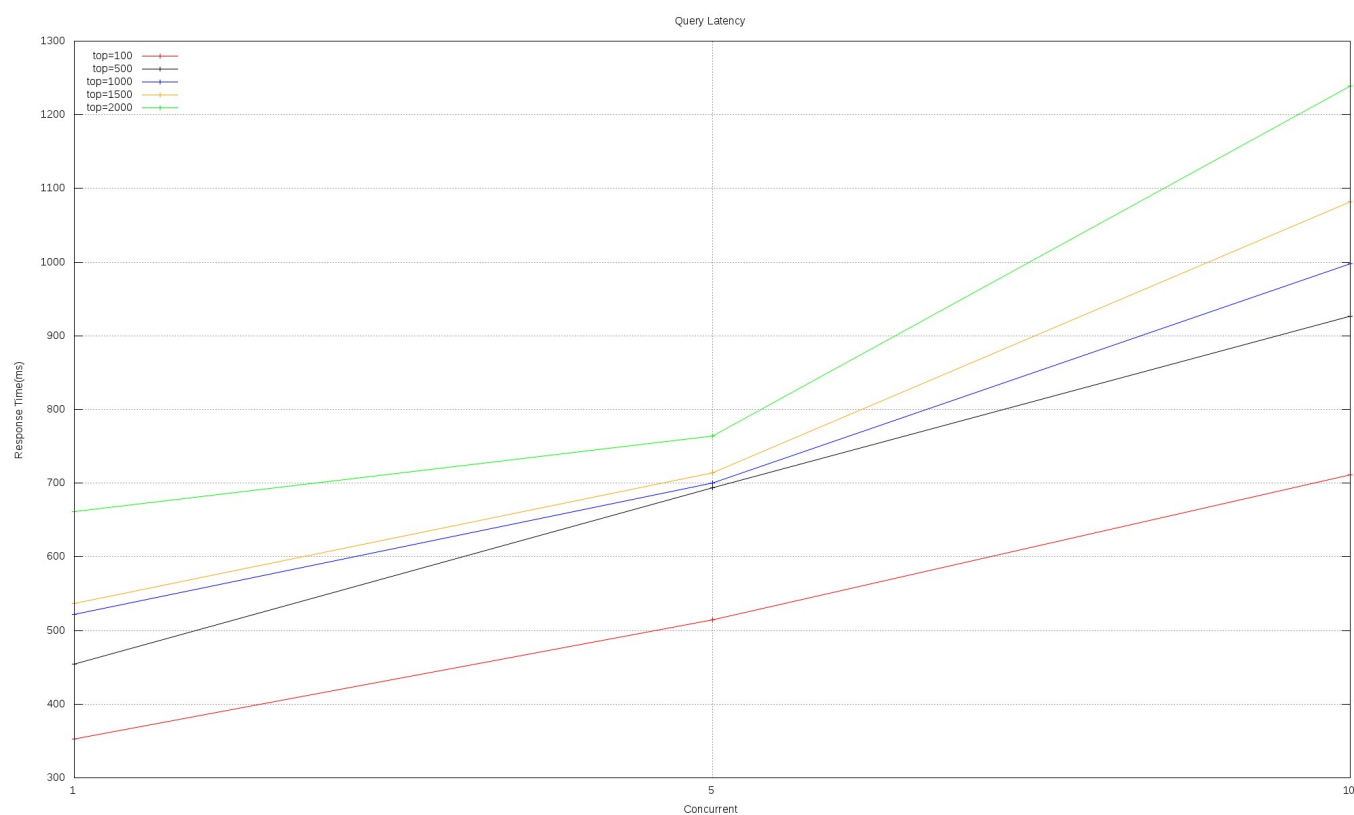
写入性能

客户端以 bulk 方式写入，每 1000 个请求构成 1 个 bulk, 并发为 4. 在 cpu/磁盘的资源没有用满的情况下，单个 data 节点的写入性能随着 shard 数增加而增加。

使用**train**这个 index 构造了 4kw 的测试集，导入耗时 30min。

查询性能

把**train** merge 为 1 个 segment, 在 4kw 的测试集下，测试不同并发下的响应时间。



横轴表示客户端的并发数，纵轴表示查询响应时间(单位为 ms)。不同的折线表示 size 取不同值的结果。

在 10 个并发的情况下执行对**train**的查询(6 个 shard), cpu 的利用率在 90%以上。

备注

1. 每个 shard 的行数不要超过 1kw，否则 index 的 merge 时间过长。
2. merge 为 1 个 segment 后，查询性能最优。