****This document explains the whole process how Harvey optimized the ES index, and how to build index and query in ES*****

****The deployment and installation of ElasticSearch and logstash and Kibana is well explained in the official website, just follow the guidance you could easily install them*****

****There is no literally code for deploying ElasticSearch and this document is simply for comparing ElasticSearch to Phoenix, the real test for the performance of ElasticSearch in real distributed system will be presented in my PPT*****

version0 : Taxi

timestamp for importing:
16:01 - 18:17 76mins
index size : 6.7G
datatype of fields:
Time => date
else => text

version1: optimal

timestamp for importing:
19:29 - 20：12 43mins
index size :6.2G, saved 0.5G disk memory
improvements:

1. all fields transform to desiered datatype
   Time => date;
   Speed, Latitude, Longitude, Direction => float
   Taxi_ID => text
2. no index on undesiered fileds
   Vacant, Speed, Direction => not indexed

version2: optimal_1

timestamp for importing:
21：32 - 22：26 54mins
index size : 5.9G, saved 0.8G disk memory

1. Vacant to boolean
   Transform the 'vacant' column in the original csv file to boolean datatype

version3: optimal_2
timestamp for importing:
12:07 - 13：05　58mins
index size : 1.4G, saved 5.3G disk memory

1. Taxi_ID field not_analyzed
   treat the whole Taxi_ID as a term
2. Doc_value disabled
   Save the disk memory at the expense of performance of aggregation and sort operations
3. _all disabled
   Disable the fuzzy search
4. norm disabled
   Disable the normalization factors for string
5. throw away 'message','host','path','@timestamp'
   Throw away some internal information storage for documents

Build index in elasticsearch:

```
PUT /optimal_2
 {
  "mappings": {
   "doc": {
    "properties": {
     "Time": {
      "type": "date",
      "doc_values": false
     },
     "Taxi ID": {
      "type": "text",
      "doc_values": false,
      "index_options": "docs"
     },
     "Latitude": {
      "type": "float",
      "doc_values": false
     },
     "Longitude": {
      "type": "float",
      "doc_values": false
     },
     "Direction": {
      "type": "text",
      "index": "false",
```

```
        "doc_values": false
       },
       "Speed": {
        "type": "float",
        "index": "false",
        "doc_values": false
       },
       "Vacant": {
        "type": "boolean",
        "index": "false",
        "doc_values": false
       }
      }
     }
    }
   }
```

Query:

ID+time, search the GPS point of taxi whose template is "1192409951" between 2017/05/21 22:47:43 to 22:51:43

```
GET /optimal_2/_search
{
  "query": {
    "bool": {
      "filter": [
        { "term":  { "Taxi ID" : "1192409951" }},
        { "range": { "Time": {
              "gt": "2017-05-21T22:47:43.000Z",
              "lt": "2017-05-21T22:51:43.000Z"
        }}}
      ]
    }
  }
}
```

Spatial + ID, search the GPS point of taxi whose template is "1192409951" within the certain region(defined by Latitude and Longitude)

```
GET /optimal_2/_search
{
  "query": {
    "bool": {
      "filter": [
        { "term":  { "Taxi ID" : "1192409951" }},
        { "range": { "Latitude": {
              "gt": "26.581883",
              "lt": "27"
        }}},
        {"range":  { "Longitude": {
              "gt": "106.581883",
              "lt": "106.7"
        }}}
      ]
    }
  }
}
```

Spatial + time, search the GPS point within the certain region between 2017/05/21 22:47:43 to 22:51:43

```
GET /optimal_2/_search
{
  "query": {
    "bool": {
      "filter": [
        { "range": { "Latitude": {
              "gt": "26.581883",
              "lt": "27"
        }}},
        {"range":  { "Longitude": {
              "gt": "106.581883",
              "lt": "106.7"
        }}},
        { "range": { "Time": {
              "gt": "2017-05-21T22:47:43.000Z",
              "lt": "2017-05-21T22:51:43.000Z"
        }}}
      ]
    }
  }
}
```

Spatial+ time + ID, search the GPS point of taxi whose template is "1192409951" within the certain region between 2017/05/21 22:47:43 to 22:51:43

```
GET /optimal_2/_search
{
  "query": {
    "bool": {
      "filter": [
       { "term":  { "Taxi ID" : "1192409951" }},
       { "range": { "Latitude": {
             "gte": "26.581883",
             "lt": "27"
       }}},
       {"range":  { "Longitude": {
             "gt": "106.581883",
             "lt": "106.7"
       }}},
       { "range": { "Time": {
             "gt": "2017-05-21T22:47:43.000Z",
             "lt": "2017-05-21T22:51:43.000Z"
       }}}
      ]
    }
  }
}
```

scroll, return all results from the query
  parameter "size" decides how many documents to be returned per page
  parameter "scroll= " decides how long the result lives

```
GET /optimal_2/_search?scroll=1m
{
  "size": 500,
  "query": {
    "bool": {
      "filter": [
       { "range": { "Latitude": {
             "gt": "26.581883",
             "lt": "27"
       }}},
       {"range":  { "Longitude": {
             "gt": "106.581883",
             "lt": "106.7"
       }}},
```

```
      { "range": { "Time": {
            "gt": "2017-05-21T22:47:43.000Z",
            "lt": "2017-05-21T22:51:43.000Z"
      }}}
     ]
   }
 }
}
```

Use the scroll ID returned from last query to retrieve the rest results

```
POST  /_search/scroll
{
   "scroll" : "1m",
   "scroll_id" :
"DXF1ZXJ5QW5kRmV0Y2gBAAAAAAAAAD4WYm9laVYtZndUQlNsdDcwakFMNjU1QQ=="
}
```