The State of Geo in Elasticsearch

Nick Knize, Elasticsearch Software Engineer
Thomas Neirynck, Kibana Visualization Area Lead
Geospatial capabilities are becoming more popular among Elasticsearch Users
Topics

Geospatial Indexing, Search, and Visualization

1. Kibana / Elastic Maps Service
2. Geo Field Mappings
3. Geo Indexing, Search, and Lucene Data Structures
4. Geo Aggregations
Kibana / Elastic Maps Service
Kibana Visualizations

Out-of-the-box visualizations for geodata in Elasticsearch

2 types

- Coordinate Maps
- Region Maps

Visualize is built on top of the Elasticsearch aggregations
Coordinate Map Visualization

Shows result of `geohash_grid` aggregations.

Shows summary of all documents that belong to a single cell.

Put location of “summarized” point in the “geo-centroid” (weighted middle). This gives a better approximate location.

The more zoomed in, the more precise the location.

Different marker-styles (bubbles, heatmap)
Example 2
Region Maps

“Choropleth maps”

Thematic maps: color intensity correspond to magnitude of metric

Shows result of terms aggregations.

“Client-side” join between the result of term aggregation and a reference shape layer.

- Polygons/Multipolygons (simple feature)
- Documents in elasticsearch need to have field that matches a property of the
Region Maps

Request traffic
Vega

Experimental feature

Vega/VegaLite is a domain language in JSON to create visualizations.

Vega has support for geographic projection.
Dashboard integration

- Use map for spatial filtering of data ...
- … and have other filters applied to your map
Elastic Maps Service
Elastic Maps Service

Reference basemapping and reference data service hosted by Elastic.

“Getting started” experience for mapping.

(1) World base map
- Base for Coordinate Map, Region Map

(2) Shape layers
- World countries, US States, Germany States, Canada Provinces, USA zip-codes
- Number of identifier fields (name in one or more languages, and ISO-identifiers)
Integrating Custom Maps
Custom base maps

- (1) Configure global base-map in kibana.yml by using Tile Map Service URL
  
tilemap.url: https://tiles.elastic.co/v2/default/{z}/{x}/{y}

- (2) Configure visualization-specific base-map using WMS (web map service)

- Requires 3rd party geo-service
  - Geoserver
  - ArcGIS Server
  - MapServer
  - ....
Custom maps examples
Custom shape layers

- geojson/topojson

- Configure in kibana.yml -> available in region maps UI

regionmap:
  includeElasticMapsService: false
  layers:
    - name: "Departments of France"
      url: "http://my.cors.enabled.server.org/france_departements.geojson"
      attribution: "INRAP"
  fields:
    - name: "department"
      description: "Full department name"
    - name: "INSEE"
      description: "INSEE numeric identifier"

- Use any web-server
  - Make sure is CORS enabled so Kibana can download the data (!)
Useful blog posts

- customization
  - https://www.elastic.co/blog/kibana-and-a-custom-tile-server-for-nhl-data
  - https://www.elastic.co/blog/custom-region-maps-in-kibana-6-0
Future
Upcoming

Elastic Maps Service

- More base layers (satellite, contours)
- Different stylesheets
- On-prem deployments

Kibana

- Elastic Maps Service integration with Vega
- No restriction on number of layers
- Support for geo_shape
  - Visualize individual documents/custom styling
  - Spatial filtering
Mappings

Geo Field Types
geo_point mapping

define

PUT crime/incidents/_mapping
{
   "properties": {
      "location": {
         "type": "geo_point",
         "ignore_malformed": true,
      }
   }
}
geo_point mapping

insert

POST crime/incidents
{
  "location" : { "lat" : 41.12, "lon" : -71.34 }
}

POST crime/incidents
{
  "location" : "41.12, -71.34"
}

POST crime/incidents
{
  "location" : [[-71.34, 41.12], [-71.32, 41.21]]
}
geo_shape mapping

```json
PUT police/precincts/_mapping
{
  "properties": {
    "coverage": {
      "type": "geo_shape",
      "ignore_malformed": false,
      "tree": "quadtree",
      "precision": "5m",
      "distance_error_pct": 0.025,
      "orientation": "ccw",
      "points_only": false
    }
  }
}
```
geo_shape mapping

```
POST police/precincts/
{
   "coverage" : {
      "type" : "polygon",
      "coordinates" : [[
         [-73.9762134, 40.7538588],
         [-73.9742356, 40.7526327],
         [-73.9656733, 40.7516774],
         [-73.9763236, 40.7521246],
         [-73.9723788, 40.7516733],
         [-73.9732423, 40.7523556],
         [-73.9762134, 40.7538588]
      ]]
   }
}
```
geo_shape mapping

insert

- Shapes are parsed using OGC and ISO standards definitions
  - OGC Simple Feature Access

- Supports the following geo_shape types
  - Point, MultiPoint
  - LineString, MultiLineString
  - Polygon (with holes), MultiPolygon (with holes)
  - Envelope (bbox)
### geo_point mapping

#### Pre 5.0

<table>
<thead>
<tr>
<th>Field</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>geohash</td>
<td>Should the geo-point also be indexed as a geohash in the <code>.geohash</code> sub-field? Defaults to <code>false</code>, unless <code>geohash_prefix</code> is <code>true</code>.</td>
</tr>
<tr>
<td>geohash_precision</td>
<td>The maximum length of the geohash to use for the <code>geohash</code> and <code>geohash_prefix</code> options.</td>
</tr>
<tr>
<td>geohash_prefix</td>
<td>Should the geo-point also be indexed as a geohash plus all its prefixes? Defaults to <code>false</code>.</td>
</tr>
<tr>
<td>ignore_malformed</td>
<td>If <code>true</code>, malformed geo-points are ignored. If <code>false</code> (default), malformed geo-points throw an exception and reject the whole document.</td>
</tr>
<tr>
<td>lat_lon</td>
<td>Should the geo-point also be indexed as <code>.lat</code> and <code>.lon</code> sub-fields? Accepts <code>true</code> and <code>false</code> (default).</td>
</tr>
<tr>
<td>precision_step</td>
<td>Controls the number of extra terms that are indexed for each lat/lon point. Defaults to 16. Ignored if <code>lat_lon</code> is <code>false</code>.</td>
</tr>
</tbody>
</table>
geo_point mapping

5.0+

geohash
Should the geo-point also be indexed as a geohash in the .geohash sub-field? Defaults to false, unless geohash_prefix is true.

geohash_precision
The maximum length of the geohash to use for the geohash and geohash_prefix options.

geohash_prefix
Should the geo-point also be indexed as a geohash plus all its prefixes? Defaults to false.

ignore_malformed
If true, malformed geo-points are ignored. If false (default), malformed geo-points throw an exception and reject the whole document.

lat_lon
Should the geo-point also be indexed as .lat and .lon sub-fields? Accepts true and false (default).

precision_step
Controls the number of extra terms that are indexed for each lat/lon point. Defaults to 16. Ignored if lat_lon is false.
## geo_shape mapping

<table>
<thead>
<tr>
<th>Option</th>
<th>Description</th>
<th>Default</th>
</tr>
</thead>
<tbody>
<tr>
<td>tree</td>
<td>Name of the PrefixTree implementation to be used: geohash for GeohashPrefixTree and quadtree for QuadPrefixTree.</td>
<td>geohash</td>
</tr>
<tr>
<td>precision</td>
<td>This parameter specifies the desired precision of the index. The value should be a number followed by an optional distance unit. Valid distance units include: in, inch, yd, yard, mi, miles, km, kilometers, m, meters, cm, centimeters, mm, millimeters.</td>
<td>meters</td>
</tr>
<tr>
<td>tree_levels</td>
<td>Maximum number of layers to be used by the PrefixTree. Defaults to the default value of the chosen PrefixTree implementation.</td>
<td>58m</td>
</tr>
<tr>
<td>strategy</td>
<td>The strategy parameter defines the approach for how to represent shapes at indexing and search time.</td>
<td>recursive</td>
</tr>
<tr>
<td>distance_error_pct</td>
<td>Used as a hint to the PrefixTree about how precise it should be.</td>
<td>0.625</td>
</tr>
<tr>
<td>orientation</td>
<td>Optionally define how to interpret vertex order for polygons / multipolygons. The default orientation (counterclockwise) complies with the OGC standard.</td>
<td>ccw</td>
</tr>
<tr>
<td>points_only</td>
<td>Setting this option to true configures the geo_shape field type for point false shapes only (NOTE: Multi-Points are not yet supported).</td>
<td></td>
</tr>
</tbody>
</table>
geo_shape mapping

### Options and Descriptions

<table>
<thead>
<tr>
<th>Option</th>
<th>Description</th>
<th>Default</th>
</tr>
</thead>
<tbody>
<tr>
<td>tree</td>
<td>Name of the PrefixTree implementation to be used: geohash for GeohashPrefixTree and quadtree for QuadPrefixTree.</td>
<td>geohash</td>
</tr>
<tr>
<td>precision</td>
<td>This parameter specifies the desired precision of the index. The value should be a number followed by an optional distance unit. Valid distance units include: in, inch, yd, yard, mi, miles, km, kilometers, m, meters, cm, centimeters, mm, millimeters.</td>
<td>meters</td>
</tr>
<tr>
<td>tree_levels</td>
<td>Maximum number of layers to be used by the PrefixTree. Defaults to the default value of the chosen PrefixTree implementation.</td>
<td>50m</td>
</tr>
<tr>
<td>strategy</td>
<td>The strategy parameter defines the approach for how to represent shapes at indexing and search time.</td>
<td>recursive</td>
</tr>
<tr>
<td>distance_error_pct</td>
<td>Used as a hint to the PrefixTree about how precise it should be.</td>
<td>0.025</td>
</tr>
<tr>
<td>orientation</td>
<td>Optionally define how to interpret vertex order for polygons / multipolygons. The default orientation (counterclockwise) complies with the OGC standard.</td>
<td>ccw</td>
</tr>
<tr>
<td>points_only</td>
<td>Setting this option to <code>true</code> configures the geo_shape field type for point. <code>false</code> shapes only (NOTE: Multi-Points are not yet supported).</td>
<td></td>
</tr>
</tbody>
</table>
Geo Indexing
geo_point indexing

2.x term/postings encoding

<table>
<thead>
<tr>
<th>term</th>
<th>postings (doc ids)</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>1, 2, 3, 4, 5</td>
</tr>
<tr>
<td>10</td>
<td>1, 2, 4</td>
</tr>
<tr>
<td>11</td>
<td>3, 5</td>
</tr>
<tr>
<td>100</td>
<td>1</td>
</tr>
<tr>
<td>101</td>
<td>2, 4</td>
</tr>
<tr>
<td>111</td>
<td>3, 5</td>
</tr>
<tr>
<td>1000</td>
<td>2</td>
</tr>
<tr>
<td>1010</td>
<td>4</td>
</tr>
<tr>
<td>1011</td>
<td>3</td>
</tr>
<tr>
<td>1110</td>
<td>3</td>
</tr>
<tr>
<td>1111</td>
<td>5</td>
</tr>
</tbody>
</table>
geo_point indexing

5.0 - “points” data structure - (Bkd-tree)
geo_point indexing

5.0 - “points” data structure - (Bkd-tree)
geo_point indexing

performance improvements
geo_shape indexing

current - terms/postings encoding

- Max tree_levels == 32 (2 bits / cell)
- distance_error_pct
  - “slop” factor to manage transient memory usage
  - % of the diagonal distance (degrees) of the shape
  - Default == 0 if precision set (2.0)
- points_only
  - optimization for points only shape index
  - short-circuits recursion
geo_shape indexing
7.0+ - “ranges” encoding (Bkd-tree)

• Dimensional Shapes represented using Minimum Bounding Ranges (MBR)
  – Ranges (1D) - Available from 5.1+ for numerics, dates, and IP (v4 and v6)
  – Rectangles (2D) - LatLonBoundingBox Available in Lucene 7.1+
  – Cubes (3D)
  – Tesseract (4D)
geo_shape indexing

performance - 1D Numerics

![Bar chart showing performance comparison between NumericField and PointField for index size and index time. The chart shows that NumericField has a lower index time of 0.31 compared to PointField's 1.](image-url)
Geo Search
geo_point search

Pre 5.0 - terms/postings encoding

- Spatial Queries
  - BoundingBox, Distance, DistanceRange, Polygon
- PRECISION_STEP controls number of query terms (must match with index)
- TwoPhaseIterator
  - Delays boundary confirmation so other query (filters, conjunctions) can pre-filter
geo_point search

5.0+ - “points” encoding (Bkd-tree)

1. Leaf cell is fully within polygon (salmon) - return all docs

2. Leaf cell crosses the boundary (gray) - two-phase check
geo_point search

5.0+ - performance improvements
geo_shape search

capabilities

• Supports the following geo_shape types
  – Point, MultiPoint
  – LineString, MultiLineString
  – Polygon (with holes), MultiPolygon (with holes)
  – Envelope (bbox)

• Supports relational queries
  – INTERSECTS, DISJOINT, WITHIN, CONTAINS
geo_shape search

current - terms/postings encoding

1. Recursively Traverse Query terms
2. Collect DocIDs from Postings based on requested relation
geo_shape search

7.0+ - “points” encoding (B-kd Tree)
geo_shape search

7.0+ - “points” encoding (B-kd Tree)
geo_shape search

1D numeric range performance

![Bar chart showing search time and heap usage comparison between NumericField and PointField. The chart indicates that NumericField has a search time of 1 and a Heap Usage of 0.15, while PointField has a search time of 0.76.]
Geo Aggregations
GeoDistance Agg
GeoDistance Agg
GeoGrid Agg

```json

{
    "aggs": {
        "crime_cells": {
            "geohash_grid": {
                "field": "location",
                "precision": 8
            }
        }
    }
}

```
GeoGrid
Agg
GeoCentroid Agg

```
"query": {
  "match": {
    "crime": "burglary"
  }
},
"aggs": {
  "towns": {
    "terms": {
      "field": "town"
    },
    "aggs": {
      "centroid": {
        "geo_centroid": {
          "field": "location"
        }
      }
    }
  }
}
```
GeoCentroid
Agg
GeoCentroid Agg
Geo Aggregations

more available, and coming soon...

- matrix_stats - (Matrix Aggs) plugin
  - kurtosis/skewness
  - variance-covariance matrix
  - pearson’s product correlation matrix

- geo_stats - Future?
  - Moran’s I - measuring spatial auto-correlation
  - Getis-Ord - spatial hot spot analysis