

Using the percolator for simple classification

Alexander Reelsen

@spinscale

alexander.reelsen@elasticsearch.com

About...

- ... Elasticsearch

Founded 2012

Offices in Mountain View, Amsterdam, London, Berlin, Phoenix

VC by Benchmark, Index Ventures & NEA

Trainings, Development/Production Support

Products: Elasticsearch, Logstash, Kibana, Marvel, Shield

- ... me

joined early 2013

interested in scalability/concurrency

Core/Shield developer, blogger, trainer, supporter, speaker

Perco... what?

- Search, but reversed
- Normal: Indexing documents & executing queries
- How about: Indexing queries and firing documents against it?

Example: Alerting

```
PUT /stocks/.percolator/sell-immediately
{
  "query": {
    "filtered": {
      "filter": {
        "bool": {
          "must": [
            { "term": { "name": "GOOG" } },
            { "range": { "value": { "gte": 1000 } } }
          ]
        }
      }
    }
  }
}
```

Example: Alerting

```
GET /stocks/stock/_percolate
{
  "doc": {
    "name": "GOOG",
    "value": 1100.52
  }
}
```

Example: Alerting

```
GET /stocks/stock/_percolate
{
  "doc": {
    "name": "GOOG",
    "value": 1100.52
  }
}
```

```
{
  ...
  "total": 1,
  "matches": [
    { "_index": "stocks", "_id": "sell-immediately" }
  ]
}
```

Example: Back-in-stock notify

```
PUT /products/.percolator/alr-XYZ
{
  "query": {
    "filtered": {
      "filter": {
        "bool": {
          "must": [
            { "term": { "id": "XYZ" } },
            { "range": { "stock": { "gt": 1 } } }
          ]
        }
      }
    }
  }
}
```

Example: Back-in-stock notify

```
GET /products/product/_percolate
```

```
{  
  "doc": {  
    "id": "XYZ",  
    "stock": 200  
  }  
}
```

Example: Back-in-stock notify

```
GET /products/product/_percolate
```

```
{  
  "doc": {  
    "id": "XYZ",  
    "stock": 200  
  }  
}
```

```
{  
  ...  
  "total": 1,  
  "matches": [  
    { "_index": "products", "_id": "alr-XYZ" }  
  ]  
}
```

Country tagging by lat/lon

- Index all countries as geoshapes as percolator queries
- Percolate a document with latitude/longitude
- Get back the country
- Index your document including the country in your index
- Aggregate/Search by country!

Enrich before index

- Document was classified (by lat/lon) before indexing!
- So, what could we use this for...?

How does it work?

- All registered queries are loaded into memory
- Each doc is indexed into in-memory index
- All queries are executed against in-memory index
- In-Memory-index gets removed
- Matched queries are returned in response

A blast from the past...

Audi 90 quattro, 170PS, ABS, eFH,
el.SP, WFS, 4WD, WR, ZV, FFB, EZ
12/90, MFA, 140000 km, HU 12/15,
5000 VB

As JSON document

```
{
  "brand" : "audi",
  "type" : "90",
  "engine" : {
    "hp" : "170",
    "cylinders" : "5",
    "capacity" : 2309
  },
  "price" : {
    "value" : 5000,
    "type" : "negotiable"
  },
  "registration" : "1990-12-01",
  "mileage" : {
    "value" : 40000,
    "unit" : "km"
  }
}
```

```
"inspection" : "2015-12-01",
"features" : [
  "anti-lockbraking-system",
  "power-windows",
  "remote-door-lock",
  "multi-functional-display",
  "power-mirrors",
  "anti-theft-protection",
  "4wd",
  "winter-tires",
  "central-locking" ]
}
```

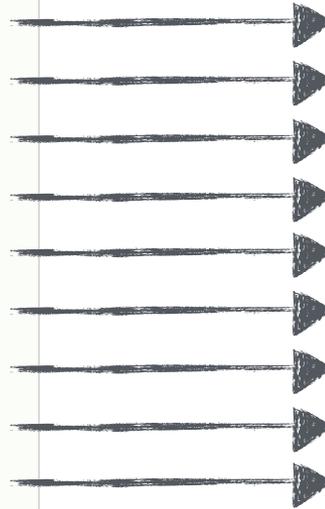
As JSON document

```
{
  "brand" : "audi",
  "type" : "90",
  "engine" : {
    "hp" : "170",
    "cylinders" : "5",
    "capacity" : 2309
  },
  "price" : {
    "value" : 5000,
    "type" : "negotiable"
  },
  "registration" : "1990-12-01",
  "mileage" : {
    "value" : 40000,
    "unit" : "km"
  }
}
```

```
"inspection" : "2015-12-01",
"features" : [
  "anti-lockbraking-system",
  "power-windows",
  "remote-door-lock",
  "multi-functional-display",
  "power-mirrors",
  "anti-theft-protection",
  "4wd",
  "winter-tires",
  "central-locking" ]
}
```

As JSON document

```
"inspection" : "2015-12-01",  
"features" : [  
  "anti-lockbraking-system",  
  "power-windows",  
  "remote-door-lock",  
  "multi-functional-display",  
  "power-mirrors",  
  "anti-theft-protection",  
  "4wd",  
  "winter-tires",  
  "central-locking" ]  
}
```



ABS
eFH
FFB
MFA
el. SP
WFS
4WD
WR
ZV

Extracting manufacturer

```
PUT /ads/.percolator/manufacturer-audi
{
  "query": {
    "match": {
      "message": "audi"
    }
  }
}
```

Extracting manufacturer

```
GET /ads/ad/_percolate
{
  "doc": {
    "message": "Audi 90 quattro, 170PS, ABS, eFH,
el.SP, WFS, 4WD, WR, ZV, FFB, EZ 12/90, MFA, 140000
km, HU 12/15, 5000 VB"
  }
}
```

```
{ ...
  "total": 1,
  "matches": [
    {
      "_index": "ads", "_id": "manufacturer-audi"
    }
  ]
}
```

Extracting features

```
PUT /ads/.percolator/feature-4wd
{
  "query": {
    "match": {
      "message": "4wd quattro 4matic awd"
    }
  }
}
```

Features with typos

```
PUT /ads/.percolator/feature-4wd
{
  "query": {
    "bool": {
      "should": [
        {
          "match": { "message": "4wd quattro 4matic awd" }
        },
        {
          "match": {
            "message": { "query": "quattro", "fuzziness": 1 }
          }
        }
      ]
    }
  }
}
```

Features with typos

```
PUT /ads/.percolator/feature-4wd
{
  "query": {
    "bool": {
      "should": [
        {
          "match": { "message": "4wd quattro 4matic awd" }
        },
        {
          "match": {
            "message": { "query": "quattro", "fuzziness": 1 }
          }
        }
      ]
    }
  }
}
```

Use-cases

- Alerting
 - Price monitoring
 - News alerts
 - Stock alerts
 - Logs
- Enrich before indexing
 - Targeted advertisement
 - Classification/Extraction
 - Wizards/Helpers

Thanks for listening!

We're hiring!

<http://elasticsearch.com/jobs>

We're helping!

<http://elasticsearch.com/support>

<http://elasticsearch.com/training>

Alexander Reelsen

@spinscale

alexander.reelsen@elasticsearch.com

The Elasticsearch logo, consisting of the word "elasticsearch." in a lowercase, green, sans-serif font.

Resources

<http://www.elasticsearch.org/blog/using-percolator-geo-tagging/>

<http://www.elasticsearch.org/blog/percolator-redesign-blog-post/>

<http://www.elasticsearch.org/guide/en/elasticsearch/reference/current/search-percolate.html>