TP4 Exploitation de données

Les Aggregations

une agrégation est une collection ou un rassemblement d'éléments connexes.

Il est possible de regrouper les documents pour pouvoir appliquer une agrégation à des fins de statistiques.

Pour cela, il faut utiliser la clé "aggs" dans les documents requêtes:

=> donner la clé où prendre les valeurs "terms : field :XXX" (identique à un GROUP BY) et donner la clé où mettre la valeur d'agrégation.

*Voici d'autres exemples d'utilisations du cadre d'agrégation :

- Temps de chargement moyen d'un site web
- Clients les plus précieux en fonction du volume de transactions
- Histogramme montrant une métrique (quantité, moyenne, somme, ...) pour les événements survenus dans des périodes générées dynamiquement
- Quantité de produits dans chaque catégorie de produits

Agrégations de métriques

Il existe plusieurs types d'agrégations :

- Agrégations de métriques
- Agrégations de compartiments
- Agrégations de pipeline

https://www.elastic.co/guide/en/elasticsearch/reference/current/search-aggregations.html

Metric aggregations

Il existe plusieurs types d'agrégations :

*** Agrégations de métriques :**

C'est des agrégations qui calculent des mesures, telles qu'une somme ou une moyenne, à partir de valeurs de champ.

Ce type se réfère principalement aux calculs mathématiques effectués sur un ensemble de documents, généralement basés sur les valeurs d'un champ numérique présent dans le document, tels que COUNT, SUM, MIN, MAX, AVERAGE etc.

Les métriques peuvent être effectuées au niveau supérieur, mais sont souvent plus utiles en tant que sous-agrégation pour calculer les valeurs d'une agrégation de compartiment.

Par défaut, cela va compter le nombre d'occurences.

```
{"aggs" : {
        "nom_de_la_cle_en_sortie" : {
                "terms" : {"field" : "nom_de_la_cle_a_grouper"}
}}}
```

https://www.elastic.co/guide/en/elasticsearch/reference/7.17/search-aggregations-metrics.html

Metric aggregations

Agrégation moyenne:

Il s'agit d'une agrégation de mesures à valeur unique qui calcule la moyenne des valeurs numériques extraites des documents agrégés.

```
POST /exams/_search?size=0
{
  "aggs": {
    "avg_grade": { "avg": { "field": "grade" } }
  }
}
```

```
{
  ...
  "aggregations": {
    "avg_grade": {
      "value": 75.0
    }
  }
}
```

=> calcule la note moyenne sur tous les documents. Le type d'agrégation est avg et le paramètre de champ définit le champ numérique des documents sur lesquels la moyenne sera calculée.

Le nom de l'agrégation (avg_value) sert également de clé permettant d'extraire le résultat de l'agrégation de la réponse renvoyée

Metric aggregations

Agrégation max:

Il s'agit d'une agrégation de mesures à valeur unique qui calcule le maximum des valeurs numériques extraites des documents agrégés.

```
POST /sales/_search?size=0
{
  "aggs": {
    "max_price": { "max": { "field": "price" } }
  }
}
```

```
{
  ...
  "aggregations": {
      "max_price": {
          "value": 200.0
      }
  }
}
```

=> calcule la valeur max sur tous les documents. Le type d'agrégation est max et le paramètre de champ définit le champ numérique des documents sur lesquels le max sera calculé.

Le nom de l'agrégation (max_price) sert de clé permettant d'extraire le résultat de l'agrégation de la réponse renvoyée

Metric aggregations

Agrégation de Cardinalité:

Une agrégation de mesures à valeur unique qui calcule un compte approximatif de valeurs distinctes.

Les valeurs peuvent être extraites de champs spécifiques du document ou générées par un script.

```
POST /sales/_search?size=0
{
  "aggs": {
    "type_count": {
      "cardinality": {
        "field": "type"
      }
    }
  }
}
```

```
{
  ...
  "aggregations": {
    "type_count": {
      "value": 3
    }
  }
}
```

Metric aggregations

Agrégation étendue des statistiques:

Une agrégation de métriques à valeurs multiples qui calcule les statistiques sur les valeurs numériques extraites des documents agrégés. Ces valeurs peuvent être extraites de champs numériques spécifiques dans les documents ou générées par un script fourni.

L'agrégation extended_stats est une version étendue de l'agrégation de statistiques, dans laquelle des mesures supplémentaires sont ajoutées, telles que sum_of_squares, variance, std_deviation et std_deviation_bounds.

```
GET /exams/_search
{
  "size": 0,
  "aggs": {
    "grades_stats": { "extended_stats": { "field": "grade" } }
  }
}
```


```
{
  ...

  "aggregations": {
    "grades_stats": {
      "count": 2,
      "min": 50.0,
      "max": 100.0,
      "avg": 75.0,
      "sum": 150.0,
      "sum_of_squares": 12500.0,
      "variance": 625.0,
      "variance_population": 625.0,
      "variance_sampling": 1250.0,
      "std_deviation": 25.0,
      "std_deviation_population": 25.0,
      "std_deviation_sampling": 35.35533905932738,
      "std_deviation_bounds": {
        "upper": 125.0,
        "lower": 25.0,
        "upper_population": 125.0,
        "lower_population": 25.0,
        "upper_sampling": 145.71067811865476,
        "lower_sampling": 4.289321881345245
      }
    }
  }
}
```

Bucket aggregations

*** Agrégations de compartiments :**

C’est des agrégations qui regroupent les documents en compartiments (buckets), également appelés bacs (bins) , en fonction des valeurs de champ, des plages ou d’autres critères du document.

Lorsque l’agrégation est effectuée, les documents sont placés dans le(s) compartiment(s) respectif(s).

De cette façon, vous pouvez diviser un ensemble de factures en plusieurs compartiments, un pour chaque client, les journaux système (logs) peuvent être divisés en « error», « warning» et « info », ou les données de performance du processeur divisées en hourly buckets.

La sortie consiste en une liste de compartiments, chacun avec une clé et un nombre de documents.

Voici quelques exemples de bucket aggregations :

Histogram Aggregation, Range Aggregation, Terms Aggregation, Filter(s) Aggregations, Geo Distance Aggregation and IP Range Aggregation.

https://www.elastic.co/guide/en/elasticsearch/reference/current/search-aggregations-bucket-terms-aggregation.html

https://www.elastic.co/guide/en/elasticsearch/reference/current/search-aggregations-bucket.html

Bucket aggregations

L'agrégation ci-dessous séparera tous les résultats d'examen par genre de l'élève, puis calculera les résultats moyens pour chaque genre.

=> la deuxième agrégation sera calculée sur le bucket plutôt que sur l'ensemble du document

```
POST exam_results*/_search
{
  "size": 0,
  "aggs": {
    "genders": {
      "terms": {
        "field": "gender"
      },
      "aggs": {
        "avg_grade": {
          "avg": {
            "field": "grades"
          }
        }
      }
    }
  }
}
```

Pipeline aggregations

*** Agrégations de pipeline :**

C’est des agrégations qui prennent des entrées provenant d’autres agrégations au lieu de documents ou de champs. Ces agrégations permettent d’agréger en fonction du résultat d’une autre agrégation plutôt que d’ensembles de documents.

En général, cette agrégation est utilisée pour rechercher le nombre moyen de documents dans un compartiment ou pour trier des compartiments en fonction d’une métrique produite par une agrégation de métriques.

https://www.elastic.co/guide/en/elasticsearch/reference/current/search-aggregations-pipeline.html

Pipeline aggregations

```
POST /_search
{
  "aggs": {
    "my_date_histo": {
      "date_histogram": {
        "field": "timestamp",
        "calendar_interval": "day"
      },
      "aggs": {
        "the_sum": {
          "sum": { "field": "lemmings" }                    (1)
        },
        "the_deriv": {
          "derivative": { "buckets_path": "the_sum" }       (2)
        }
      }
    }
  }
}
```

(1) The metric is called `"the_sum"`

(2) The `buckets_path` refers to the metric via a relative path `"the_sum"`

Pipeline aggregations

```
POST /_search
{
  "aggs": {
    "sales_per_month": {
      "date_histogram": {
        "field": "date",
        "calendar_interval": "month"
      },
      "aggs": {
        "sales": {
          "sum": {
            "field": "price"
          }
        }
      }
    },
    "max_monthly_sales": {
      "max_bucket": {
        "buckets_path": "sales_per_month>sales" (1)
      }
    }
  }
}
```

(1) `buckets_path` instructs this max_bucket aggregation that we want the maximum value of the `sales` aggregation in the `sales_per_month` date histogram.

Les Aggregations: Synatxe

```
{"aggs" : {
        "nom_de_la_cle_en_sortie" : {
                "terms" : {"field" : "nom_de_la_cle_a_grouper"}
}}}
```

Attention, cette méthode va décomposer les valeurs de la clé ”nom_de_la_cle_a_grouper”. Ainsi, un texte se verra décomposer en ensemble de mots pour être ensuite regroupés (par mot).

Il est possible de filtrer les documents avant l’agrégat. Il suffit de faire une requête classique ”query”

```
{"query" :{
                "match" : {"cle_a_filtrer" : "valeur"}
        }
 ,"aggs" : {
                "cle_de_sortie" : {
                        "avg" : {"field" : "cle_a_grouper"}
        }}}
```

Les Aggregations: Syntaxe

```
GET movies/_search
{"query":
  {"match": {
    "fields.actors": "angelina jolie"
}},
"aggs": {
  "last_one": {
    "max": {
      "field": "fields.year"
    }
  }
}
}
```

```
#! Elasticsearch built-in security features are not en
  ://www.elastic.co/guide/en/elasticsearch/reference/7
{
  "took" : 7,
  "timed_out" : false,
  "_shards" : {
    "total" : 1,
    "successful" : 1,
    "skipped" : 0,
    "failed" : 0
  },
  "hits" : {...},
  "aggregations" : {
    "last_one" : {
      "value" : 2015.0
    }
  }
}
```

Nesting aggregations

*** Aggrégation imbriquées:**

Il est possible d'imbriquer des agrégations les unes dans les autres ,pour grouper par type ,(rien à voir avec les champs imbriqués), de manière à diviser les buckets en sub buckets ,ou de calculer des métriques à partir des sous-compartiments.

```
{"aggs" : {
        "nom_groupe" : {
                "terms" : {
                        "field" : "cle_du_group_by"
                },
                "aggs" : {
                        "nom_sous_groupe" : {
                                "avg" : {"field" : "cle_pour_la_moyenne"}
                        }}
                }
        }}
```

Le tri fonctionne de la même manière avec la clé "order" en utilisant le nom de la clé à trier.

"order" : { "cle_pour_la_moyenne" : "desc" }

Nesting aggregations

```
POST products/_search
{
  "size": 0,
  "aggs": {
    "by-country": {
      "terms": {
        "field": "country"
      },
      "aggs": {
        "stock": {
          "sum": {
            "field": "qty"
          }
        }
      }
    }
  }
}
```

- Chaque agrégation (ou sous-agrégation) a un nom (by-country et stock,).
- Nous avons défini la taille des résultats (size) à 0, ce qui signifie que nous n'obtenons aucun résultat dans la réponse (c'est recommandé).
- Nous n'avons utilisé , dans cet exemple, que les types d'agrégation terms (bucket aggregation) et sum (metric aggregation)
- On a utilisé une sous-agrégation, l'agrégation by-country crée des buckets (groupes) de résultats, puis l'agrégation de stock donne une métrique pour chaque compartiment.
-On peut imbriquer plusieurs agrégations de compartiments avant d'exécuter à la fin (et éventuellement) une agrégation de métriques dessus.

Les Aggregations: Syntaxe

Le tri fonctionne de la même manière avec la clé ”order” en utilisant le nom de la clé à trier.

"order" : { "cle_pour_la_moyenne" : "desc" }

Il est également possible de grouper par plages de valeur plutot que les valeurs du document :

```
{"aggs" : {
        "nom_cle_sortie" : {
                "range" : {
                        "field" : "cle_ou_prendre_la_valeur",
                                "ranges" : [
                                        {"to" : "valeur_max"},
                                        {"from" : "valeur_min", "to" : "valeur_max"},
                                        {"from" : "valeur_min"}
                                ]
                        }
                }}}
```

Performances

Les agrégations sont généralement effectuées dans la mémoire RAM et nécessitent une structure d'accès au document différente de celle d'une requête de recherche obtenue à partir de l'index inversé, il est donc important de prendre en compte l'impact des performances lors des agrégations.

Voici ce qu'on peut prendre en considération :

*** Nobmre de buckets:**

à contrôler par le paramètre « size » dans une agrégation terms, ou aussi via «calendar interval » dans un date histogram. lorsqu'on a des agrégations de compartiments imbriquées à plusieurs niveaux, le nombre total de buckets sera multiplié pour chaque niveau d'agrégation.

*** Nombre de documents:**

Lors de l'exécution d'une agrégation, il est préférable (si possible) d'ajuster la requête afin que l'agrégation ne soit effectuée que sur un ensemble restreint de documents qui vous intéressent, au lieu d'utiliser une requête match_all. Cela réduira la mémoire requise pour exécuter l'agrégation.

Performances

*** FieldData:**

En règle générale, les agrégations doivent toujours être exécutées sur des champs de type mot-clé (keyword type fields), et non sur du analysed text. Il est possible d’exécuter sur du texte analysé en utilisant le paramètre de mapping« fielddata »:"true », mais cela consomme beaucoup de mémoire et doit être évité si possible.

Fielddata

*** FieldData:**

Les champs de texte peuvent faire l'objet d'une recherche par défaut, mais par défaut ne sont pas disponibles pour les agrégations, le tri ou les scripts. Si vous essayez de trier, d'agréger ou d'accéder à des valeurs à partir d'un script sur un champ de texte, vous aurez une exception indiquant=>

Fielddata est désactivé par défaut sur les champs de texte. Set fielddata=true on your_field_name in order to load fielddata in memory by uninverting the inverted index.

Notez que cela peut cependant utiliser beaucoup de mémoire.

FieldData est le seul moyen d'accéder aux jetons analysés à partir d'un champ de texte dans des agrégations, des tris ou des scripts.

Par exemple, un champ de texte comme New York serait analysé comme new et york. Pour agréger sur ces tokens, il faut définir un fieldData

Fielddata

Enabling fielddata on `text` fields

You can enable fielddata on an existing `text` field using the update mapping API as follows:

```
PUT my-index-000001/_mapping
{
  "properties": {
    "my_field": { (1)
      "type":     "text",
      "fielddata": true
    }
  }
}
```

(1) The mapping that you specify for `my_field` should consist of the existing mapping for that field, plus the `fielddata` parameter.

https://www.elastic.co/guide/en/elasticsearch/reference/current/indices-put-mapping.html

mapping API

Vérifier le contenu inséré via l'URL elasticsearch :

http://localhost:9200/movies2/_mappings

http://localhost:9200/movies2/?pretty

http://localhost:9200/movies2/_search

http://localhost:9200/_cat/indices?v

```
{
    "mappings" : {
   "properties" : {
          "fields" : {
            "properties" : {
              "actors" : {
                "type" : "text",
                "fields" : {
                  "raw" : {
                    "type" : "keyword",
                    "index" : false
                  }
                }
              },
              "directors" : {
                "type" : "text",
                "fields" : {
                  "raw" : {
                    "type" : "keyword",
                    "index" : false
                  }
                }
              }
        }}
}
```

curl -XPUT -H "Content-Type: application/json" localhost:9200/movies2 -d @mapping.json

curl -XPUT -H "Content-Type: application/json" localhost:9200/_bulk --data-binary @/movies.json

Application

Vous êtes mnt prêt pour le TP4 => voir Notebook