Elasticsearch Aggregations with Parent-Child Relationships (https://qbox.io/blog/elasticsearch-aggregations-parent-child-relationships?utm_source=qbox.io&utm_medium=article&utm_campaign=elasticsearch-aggregations-parent-child-relationships)

Posted by Adam Vanderbush (https://qbox.io/blog/author/?utm_source=qbox.io&utm_medium=article&utm_campaign=elasticsearch-aggregations-parent-child-relationships) October 19, 2017

(https://qbox.io/blog/author/?utm_source=qbox.io&utm_medium=article&utm_campaign=elasticsearch-aggregations-parent-child-relationships)

In the past few articles, we have focused on indexing and searching parent-child relationships in elasticsearch. The parent-child functionality allows us to associate one document type with another, in a one-to-many relationship, or one parent to many children. In this tutorial, we continue with parent-child aggregations in elasticsearch.

*For this post, we will be using hosted Elasticsearch on Qbox.io. You can sign up or **launch your cluster here (https://qbox.io/signup?utm_source=blog&utm_campaign=tutorial&utm_term=launch_your_cluster&utm_medium=article)**, or click "Get Started" in the header navigation. If you need help setting up, refer to "**Provisioning a Qbox Elasticsearch Cluster. (https://qbox.io/blog/provisioning-a-qbox-elasticsearch-cluster?utm_source=tutorial&utm_term=provision&utm_medium=article&utm_campaign=index-attachments-files-elasticsearch-mapper)**"*

The advantages that parent-child has over nested objects are as follows:

- The parent document can be updated without reindexing the children.

- Child documents can be added, changed, or deleted without affecting either the parent or other children. This is especially useful when child documents are large in number and need to be added or changed frequently.

- Child documents can be returned as the results of a search request.

Tutorial

Let's index a few parent documents similar to our previous post:

```
curl -XPOST 'ES_HOST:ES_PORT/academy/location/_bulk' -d
'{ "index": { "_id": "newyork" }}
{ "name": "Manhattan Academy", "state": "New York State", "country": "USA" }
{ "index": { "_id": "london" }}
{ "name": "London Central", "state": "England", "country": "UK" }
{ "index": { "_id": "tokyo" }}
{ "name": "Tokyo Academy", "state": "The Greater Tokyo Area", "country": "Japa
```

Let's associate a few children (**player**) to our parent documents:

```
curl -XPOST 'ES_HOST:ES_PORT/academy/player/_bulk' -d
'{ "index": { "_id": 2, "parent": "london" }}
{ "name": "John Doe", "dob": "1998-07-18", "sport": "volleyball" }
{ "index": { "_id": 3, "parent": "newyork" }}
{ "name": "WIlliam Smith", "dob": "1996-11-07", "sport": "basketball" }
{ "index": { "_id": 4, "parent": "tokyo" }}
{ "name": "John Henry", "dob": "1995-07-15", "sport": "billiards" }'
```

Children Aggregation

Children Aggregation is special single bucket aggregation that enables aggregating from buckets on parent document types to buckets on child documents.

This aggregation relies on the `_parent` field in the mapping. This aggregation has a single option i.e., `type` which is to indicate the child type that the buckets in the parent space should be mapped to.

Interested in Containers? Check Out Our Enterprise Kubernetes Support (https://supergiant.io/solutions)

Parent-child supports a children aggregation as a direct analog to the nested aggregation discussed in **Nested Aggregations (https://qbox.io/blog/elasticsearch-aggregations-nested-documents-tutorial?utm_source=qbox.io&utm_medium=article&utm_campaign=elasticsearch-aggregations-parent-child-relationships)**. A parent aggregation (the equivalent of `reverse_nested`) is not supported. This example demonstrates how we could determine the favourite sports of our players by country.

Here, the country field is in the location documents. The children aggregation joins the parent documents with their associated children of type player. The sport field is from the player child documents.

```
curl -XGET 'ES_HOST:ES_PORT/academy/location/_search' -d '{
  "size" : 0,
  "aggs": {
    "country": {
      "terms": {
        "field": "country"
      },
      "aggs": {
        "players": {
          "children": {
            "type": "player"
          },
          "aggs": {
            "sport": {
              "terms": {
                "field": "sport"
              }
            }
          }
        }
      }
    }
  }
}'
```

Let's say we have an index of queries and responses to questions posted in an online forum. The response type has the following `_parent` field in the mapping:

```
curl -XPUT 'ES_HOST:ES_PORT/child?pretty' -H 'Content-Type: application/json'
    "mappings": {
        "response" : {
            "_parent" : {
                "type" : "query"
            }
        }
    }
}'
```

The query typed document contain a tag field and the response typed documents contain an user field. With the children aggregation the tag buckets can be mapped to the user buckets in a single request even though the two fields exist in two different kinds of documents.

An example of a query typed document:

```
curl -XPUT 'ES_HOST:ES_PORT/child/query/1?pretty' -H 'Content-Type: applicatio
    "body": "What's the best cloud solution provider for fully managed, hosted
    "title": "We have a requirement to host our enterprise Search Application o
    "tags": [
        "elasticsearch",
        "hosted",
        "fully-managed"
    ]
}'
```

A few examples of response typed documents:

```
curl -XPUT 'ES_HOST:ES_PORT/child/response/1?parent=1&refresh&pretty' -H 'Cont
    "user": {
        "location": "New York, United STates",
        "profile_name": "Fred",
        "id": 1
    },
    "body": "Qbox makes it easy for us to provision an Elasticsearch cluster wi
    "creation_date": "2017-09-12T12:25:21.030"
}'
```

```
curl -XPUT 'ES_HOST:ES_PORT/child/response/2?parent=1&refresh&pretty' -H 'Cont
    "user": {
        "location": "London, United Kingdom",
        "profile_name": "John",
        "id": 2
    },
    "body": "Use Qbox. It provides Free 24/7 support and maintenance for every
    "creation_date": "2017-01-12T10:23:31.030"
}'
```

The following request can be built that connects the two together:

```
curl -XPOST 'ES_HOST:ES_PORT/child/_search?size=0&pretty' -H 'Content-Type: ap
 "aggs": {
    "top-tags": {
      "terms": {
        "field": "tags.keyword",
        "size": 10
      },
      "aggs": {
        "to-responses": {
          "children": {
            "type" : "response"
          },
          "aggs": {
            "top-users": {
              "terms": {
                "field": "user.profile_name.keyword",
                "size": 10
              }
            }
          }
        }
      }
    }
  }
}'
```

The above example returns the top queries tags and per tag the top response users. Here, the top-tags aggreagtion determine the number of query documents with the tag **elasticsearch**, **managed**, etc.

The to-responses aggregation tells about response documents that are related to question documents with the tag **elasticsearch**, **managed**, etc.

```json
{
    "timed_out": false,
    "took": 32,
    "_shards": {
        "total": 5,
        "successful": 5,
        "skipped" : 0,
        "failed": 0
    },
    "hits": {
        "total": 3,
        "max_score": 0.0,
        "hits": []
    },
    "aggregations": {
        "top-tags": {
            "doc_count_error_upper_bound": 0,
            "sum_other_doc_count": 0,
            "buckets": [
                {
                    "key": "elasticsearch",
                    "doc_count": 1,
                    "to-responses": {
                        "doc_count": 2,
                        "top-users": {
                            "doc_count_error_upper_bound": 0,
                            "sum_other_doc_count": 0,
                            "buckets": [
                                {
                                    "key": "Fred",
                                    "doc_count": 1
                                },
                                {
                                    "key": "John",
                                    "doc_count": 1
                                }
                            ]
                        }
                    }
                },
                {
                    "key": "hosted",
                    "doc_count": 1,
                    "to-responses": {
                        "doc_count": 2,
                        "top-users": {
                            "doc_count_error_upper_bound": 0,
                            "sum_other_doc_count": 0,
                            "buckets": [
                                {
                                    "key": "Fred",
                                    "doc_count": 1
                                },
                                {
                                    "key": "John",
                                    "doc_count": 1
                                }
                            ]
                        }
                    }
                },
```

```
    {
      "key": "fully-managed",
      "doc_count": 1,
      "to-responses": {
        "doc_count": 2,
        "top-users": {
          "doc_count_error_upper_bound": 0,
          "sum_other_doc_count": 0,
          "buckets": [
            {
              "key": "Fred",
              "doc_count": 1
            },
            {
              "key": "John",
              "doc_count": 1
            }
          ]
        }
      }
    }
  ]
}
}
```

Give it a Whirl!

It's easy to spin up a standard hosted Elasticsearch cluster on any of our 47 Rackspace, Softlayer, or Amazon data centers. And you can now provision your own **AWS Credits on Qbox Private Hosted Elasticsearch (https://qbox.io/elasticsearch-on-aws? utm_source=qbox.io&utm_medium=article&utm_campaign=elasticsearch-aggregations-parent-child-relationships)**.

Questions? Drop us a note, and we'll get you a prompt response.

Not yet enjoying the benefits of a hosted ELK-stack enterprise search on Qbox? We invite you to **create an account today (https://qbox.io/signup? utm_source=blog&utm_campaign=tutorial&utm_term=launch_your_cluster&utm_medium=article)** and discover how easy it is to manage and scale your Elasticsearch environment in our cloud hosting service.

Get Started with Qbox

Qbox Hosted Elasticsearch

If You Found this Helpful, Share!

Categorized under:

**Aggregations (https://qbox.io/blog/tag/aggregations?
utm_source=qbox.io&utm_medium=article&utm_campaign=elasticsearch-
aggregations-parent-child-relationships)**

**Elasticsearch (https://qbox.io/blog/tag/elasticsearch?
utm_source=qbox.io&utm_medium=article&utm_campaign=elasticsearch-
aggregations-parent-child-relationships)**

Not yet enjoying the benefits of a hosted ELK-stack enterprise search on Qbox? Discover how easy it is to manage and scale your Elasticsearch environment.

Get Started (https://qbox.io/signup)

5 minutes to get started