

Elasticsearch+Kibana/Kibi as QA tools



Iwona Sputowska and Jihyun Bhom

02.11.2016

Outline

Presentation of Elasticsearch+Kibana/Kibi as a possible tools for QA analysis.

Plan:

1. What are Elasticsearch, Kibana and Kibi?
2. How dose it work: The Example;
3. ES Parent-Child Relationship;
4. Summary.

The tools: What are Elasticsearch, Kibana and Kibi ?

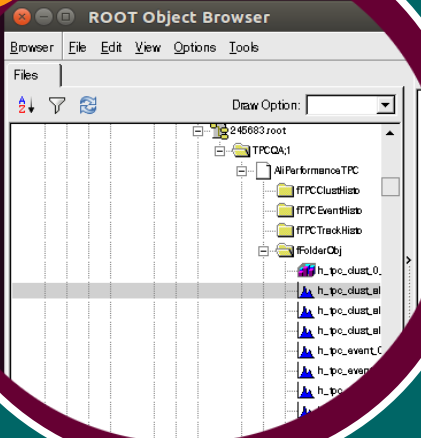
Visualisation:



How dose it work?

1.

File.root



Conversion of the
ROOT data to JSON files.

2.

```
{
  "mappings": {
    "run_245692": {
      "properties": {
        "run": { "type": "integer" },
        "mean": { "type": "float" },
        "rms": { "type": "float" },
        "alert1": { "type": "float" },
        "alert2": { "type": "float" }
      }
    }
  }
}
```

```
{ "index": { "_id": "7" } }
{
  "run": "245738",
  "mean": "621.000000",
  "rms": "604.218933",
  "alert1": "948.224304",
  "alert2": "964.000549"
}
```

data.json

Mapping

3.

Elasticsearch

+

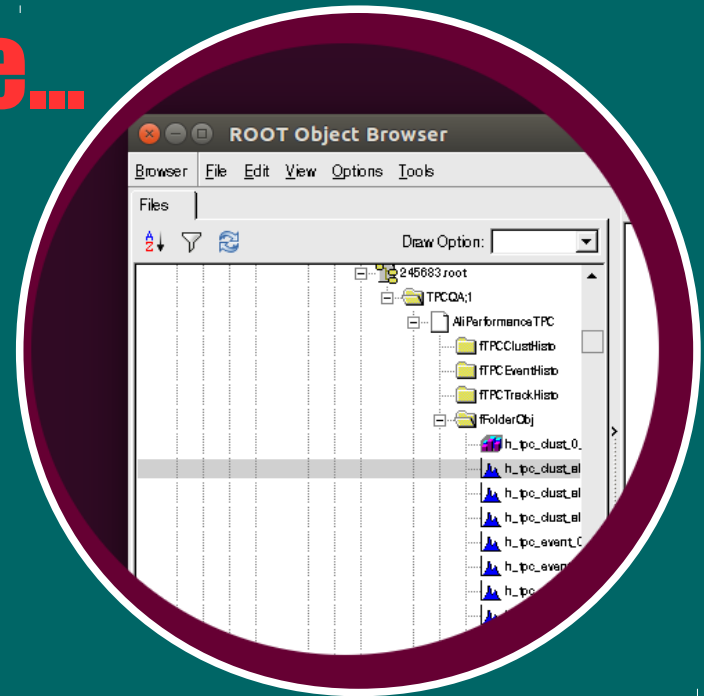
Kibana/Kibi
to visualisation

MONITORING

QA Alerts!

How does it work?: An Example...

- Production: LHC15o_pass1;
- Number of runs: 27;
- ROOT file: QAresults.root;
- Information → run-by-run trending:
 - of mean of minimum bias multiplicity distribution;



ROOT data to JSON files:

```
{
  "mappings": {
    "multiplicity": {
      "properties": {
        "run": { "type": "integer" },
        "mean": { "type": "float" },
        "rms": { "type": "float" },
        "alert1": { "type": "float" },
        "alert2": { "type": "float" }
      }
    }
  }
}
```

Mapping

```
{ "index": { "_id": "7" } }
{
  "run": "245738",
  "mean": "621.000000",
  "rms": "604.218933",
  "alert1": "948.224304",
  "alert2": "964.000549"
}
```

Document

fields

_type:multiplicity

curl -XPOST 'localhost:9200/_index -d @mapping.json

Explore the data...

KIBANA: <http://YOURDOMAIN.com:5601>

KIBI: <http://YOURDOMAIN.com:5606>

The screenshot shows the Kibi interface with a search query `run_*` entered in the query bar. The results list shows several entries with fields like `run`, `mean`, `rms`, `alert1`, `alert2`, `_id`, and `_type`. A red arrow points to the query bar, which is labeled "query bar". A red circle with the number "1." is positioned above the query bar.

run	mean	rms	alert1	alert2	_id	_type
245,683	956.486	1,061.472	948.224	964.001	0	multiplicity
245,683	0.834	0.109	0.828	0.849	0	clusters
245,829	957.223	1,067.627	948.224	964.001	14	multiplicity
245,829						
245,952	957.685	1,072.186	948.224	964.001	19	multiplicity

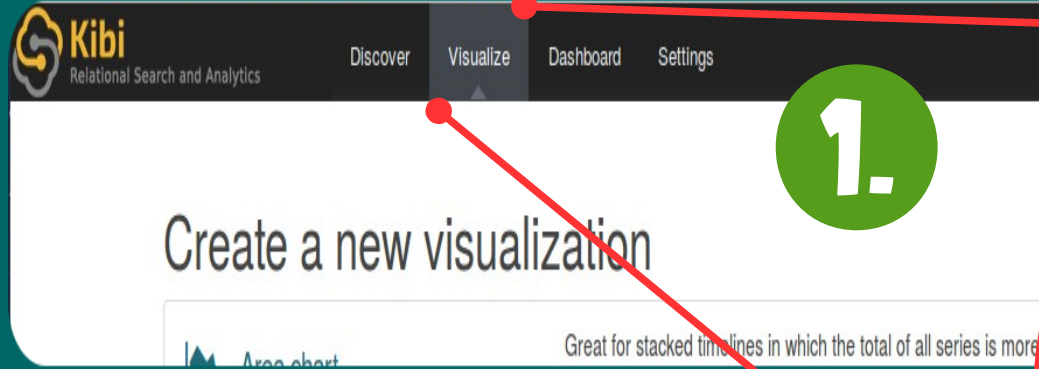
2.

The screenshot shows the Kibi interface with a search query `_type: multiplicity` entered in the query bar. The results list shows several entries with fields like `run`, `mean`, `rms`, `alert1`, `alert2`, `_id`, and `_type`. A red circle with the number "2." is positioned above the query bar. An orange circle highlights the "Save Search" dialog box, which contains the text "MULTIP" and a "Save" button. A red circle with the number "3." is positioned above the "Save" button.

run	mean	rms	alert1	alert2	_id	_type
245,683	956.486	1,061.472	948.224	964.001	0	multiplicity
245,829	957.223	1,067.627	948.224	964.001	14	multiplicity
245,952	957.685	1,072.186	948.224	964.001	19	multiplicity

5

Visualization of the data...



Kibi
Relational Search and Analytics









Discover Visualize Dashboard Settings

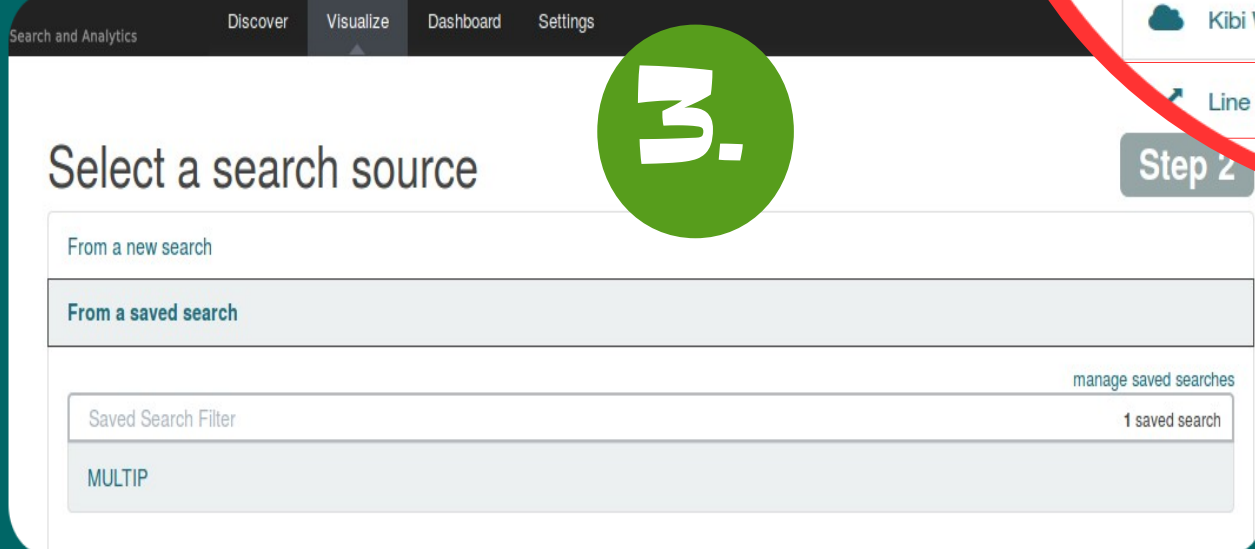
Create a new visualization

Area chart
Great for stacked timelines in which the total of all series is more

1.

Create a new visualization

-  **Area chart**
Great for stacked timelines in which the total of all series is more relative change of unrelated data points as change
-  **Data table**
The data table provides a detailed breakdown, in tabular form, of other charts by clicking grey bar at the bottom of the chart
-  **Enhanced search results**
Display search results - just like "searches" - but allows for more selections.
-  **Kibi Query Viewer**
Your SQL/SPARQL queries results here (which can be paginated)
-  **Kibi Relational filter**
Relational widget displays buttons which allow user to select related data
-  **Kibi Timeline**
Timeline widget for visualization of events
-  **Kibi Word Cloud**
Visualize a word cloud from high frequency terms
-  **Line chart**
Often the best chart for high density data as individual data points can be misleading.



Search and Analytics

Discover Visualize Dashboard Settings

Select a search source

From a new search

From a saved search

manage saved searches

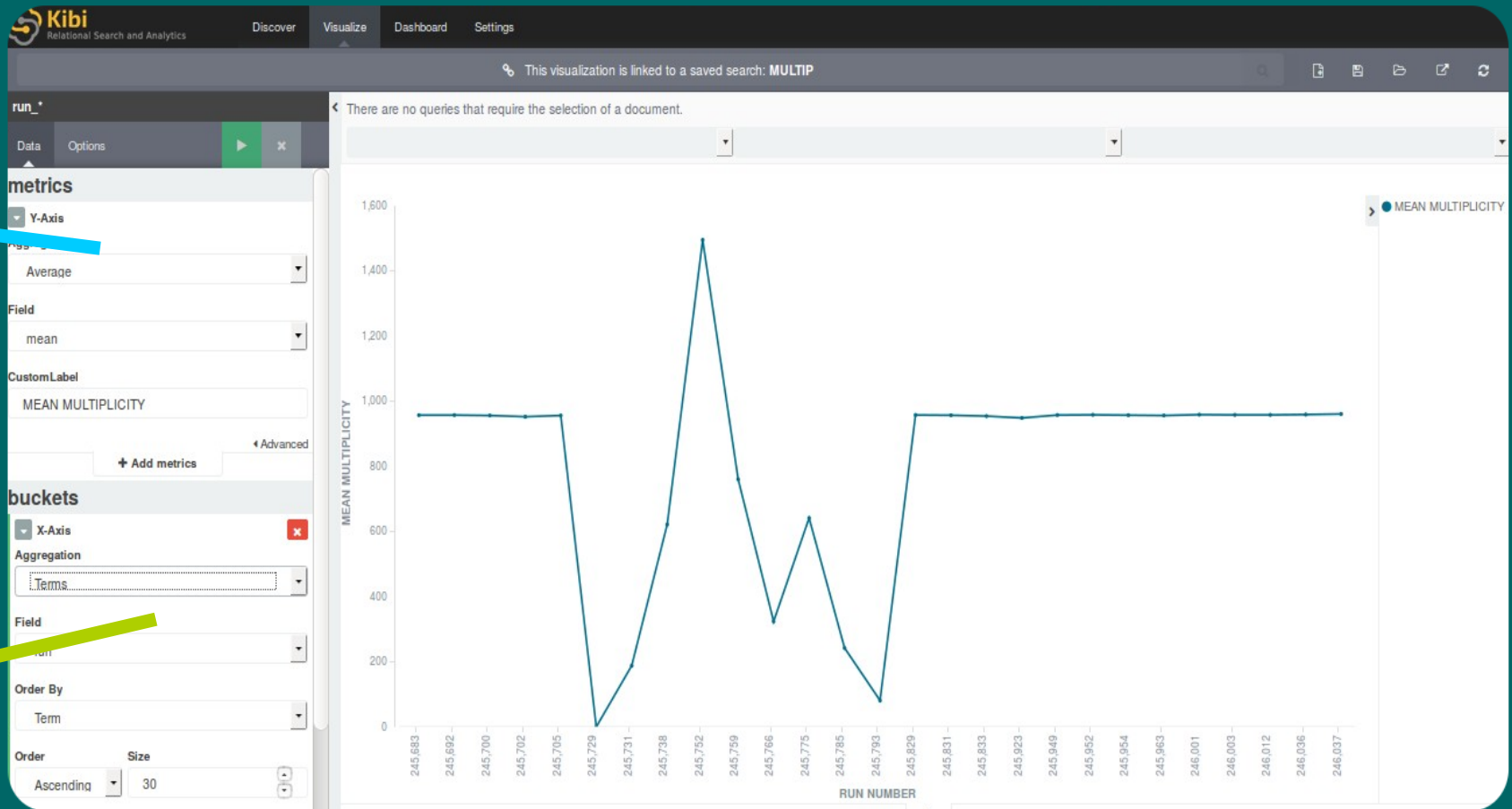
Saved Search Filter 1 saved search

MULTIP

Step 2

3.

Visualization of the data...



Y Axis:
mean MB
multiplicity

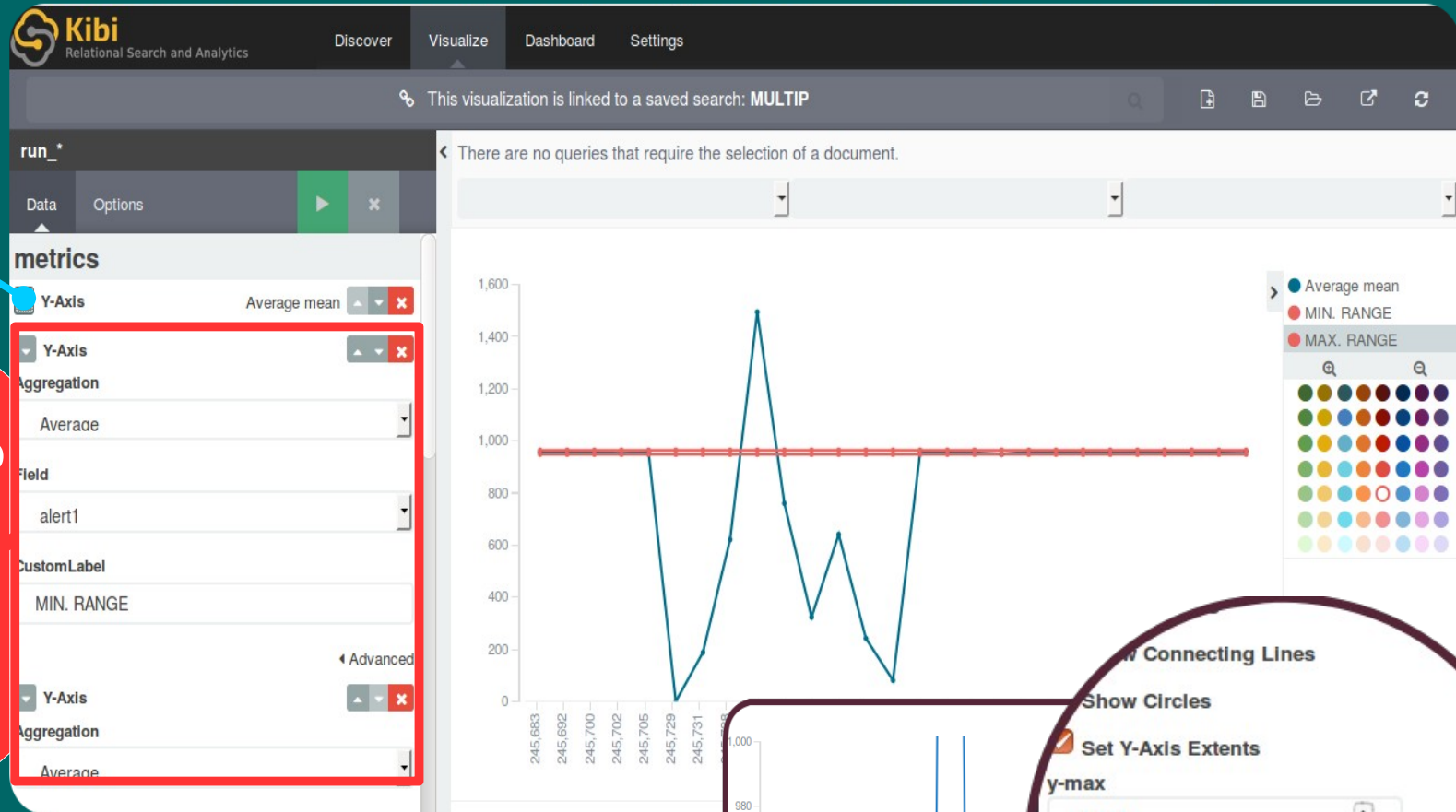
X Axis
run number

Visualization of the data: **Some Alerts!**

Y Axis:
mean MB multiplicity.

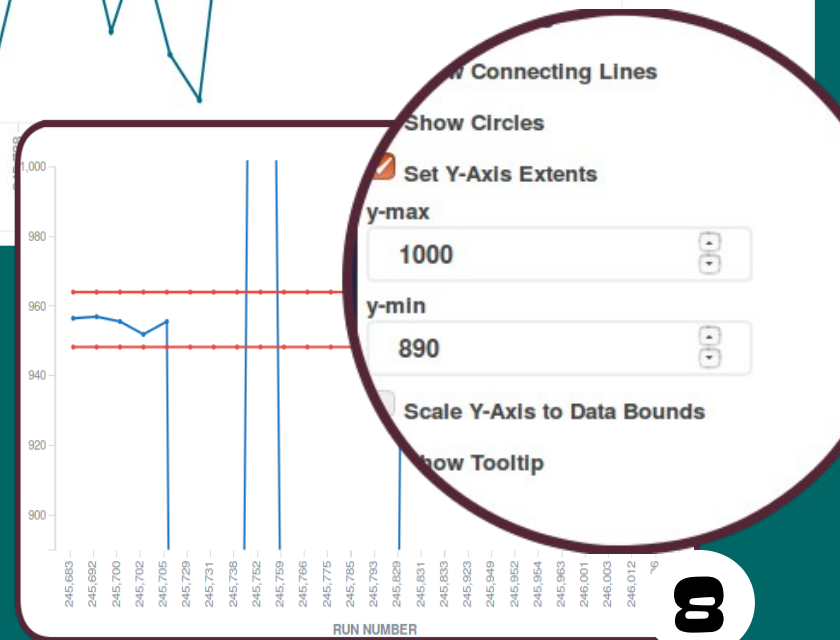
1.

Superimposed two alerts selection criteria arbitrary implemented in the JSON files.



2.

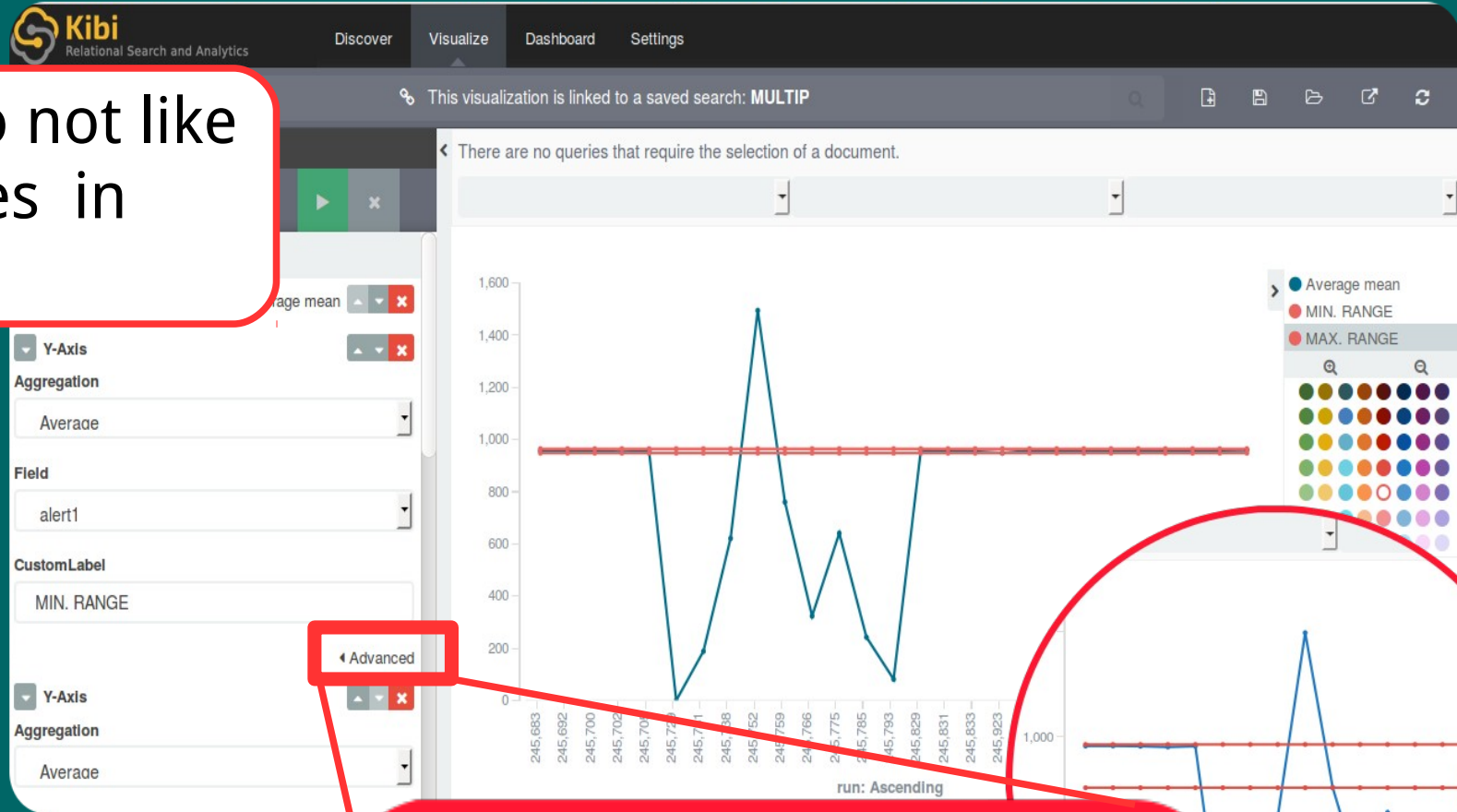
One can easily change a visualization properties.



Visualization of the data: **Some Alerts!**

1.

What if one do not like the alert ranges in JSON file?!



...they can be easily changed using **Kibana/Kibi script options...**

JSON Input ⓘ

```
{"script": "doc[alert1].value*0.8", "lang": "groovy"}
```

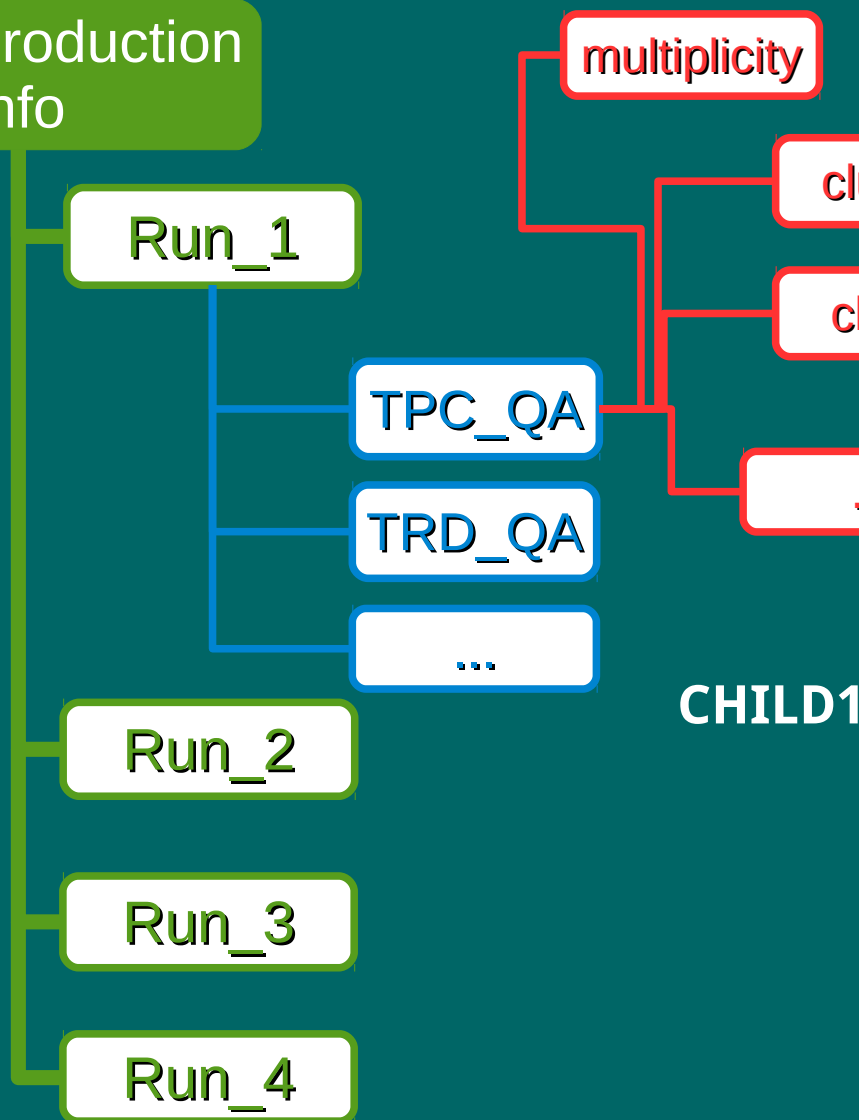
2.

3.

Parent-Child Relationship

The consider QA document structure in ES:

Index: production info



```
{
  "mappings": {
    "lhc15o_pass1": {
      "properties": {
        "run": { "type": "integer" }
      }
    },
    "tpc_qa": {
      "_parent": {
        "type": "lhc15o_pass1"
      },
      "properties": {
        "tpc_info1": { "type": "float" },
        "tpc_info2": { "type": "integer" }
      }
    },
    "trd_qa": {
      "_parent": {
        "type": "LHC15o_pass1"
      },
      "properties": {
        "trd_info1": { "type": "float" },
        "trd_info2": { "type": "float" }
      }
    }
  }
}
```

PARENT

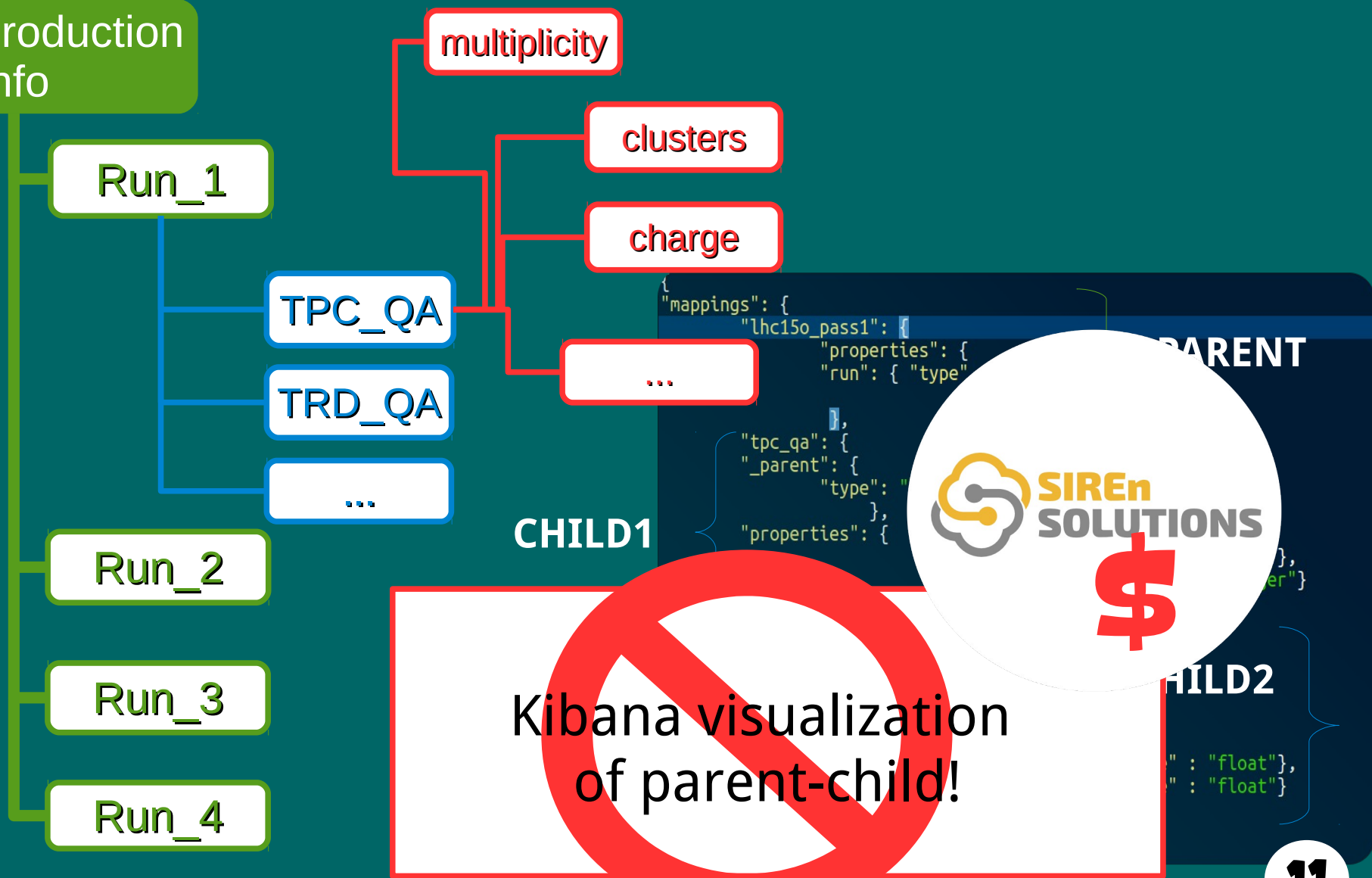
CHILD1

CHILD2

Parent-Child Relationship

The consider QA document structure in ES:

Index: production info


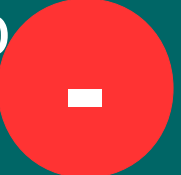


~~Kibana visualization of parent-child!~~



Summary

1. ES/Kibana/Kibi Pros and Cones:

- ES/Kibana/Kibi: available, open source tools; 
- ES fast search engine;
- Kibana/Kibi -"user friendly" visualization tools;
- Kibana- no visualization of correlations between two documents; 
- Kibi – correlations available but not in open source :(

2. **TO DO:** Test ES/Kibana/Kibi to read large data/statistic, we are going to use Logstash.

THANK YOU!