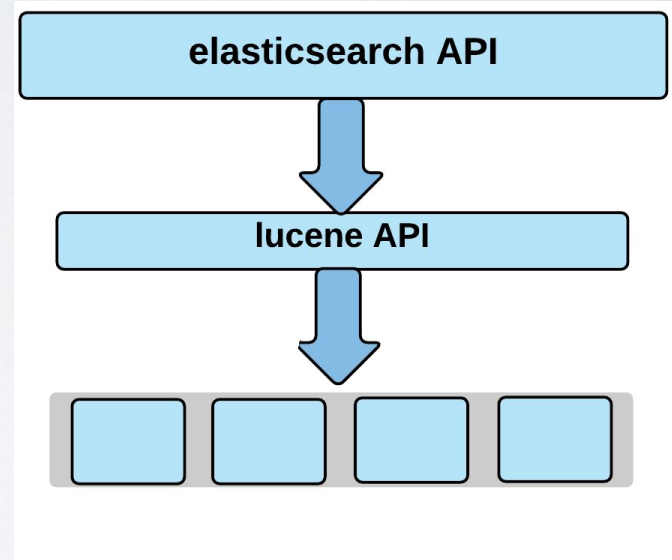


ELASTICSEARCH

Tech Talk Fridays

LUCENE

- Elasticsearch is a distributed web server written on Lucene. It's accessed via HTTP requests.
- Lucene is a java library. It can be included in projects. It can be used via function calls.



ALGORITHMS IN LUCENE

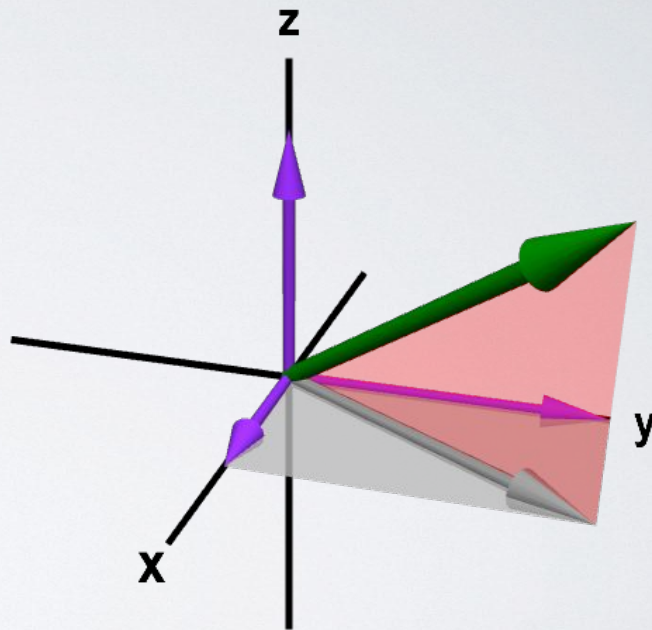
- `TFIDFSimilarity` : TF-IDF-Similarity
- Levenshtein Algorithm

TF-IDF-SIMILARITY (1/6)

- Lucene does a one-zero, true/false, boolean filtering of documents.
- Filtered documents are scored in using TF-IDF values as weights.

TF-IDF-SIMILARITY (2/6)

- This scoring of documents is called Vector Space Model, because each index term is a vector and TF-IDF value is the weight.



TF-IDF-SIMILARITY (3/6)

- TF varies with the number of occurrences of a term in the given document.

Now, why should the **apple**, in particular, be known as the "King" of all fruits? The answer is: because this fruit has so many virtues. An **apple** is beautiful, tasty and has a fine fragrance. It can be grown almost everywhere, and it keeps well.?

On the outside it is a colorful fruit. It comes in various colors from green to red to gold. In taste an **apple** is tasty and juicy, and has various tastes from sour to sweet. One can eat an **apple** raw, cooked, or baked, and it can be used in various dishes, such as **apple** tzimmes, **apple** compote, **apple** pie, **apple** strudel, etc.

TF-IDF-SIMILARITY (4/6)

- IDF varies with the number of documents containing the given term.

Now, why should the **apple**, in particular, be known as the "King" of all fruits? The answer is: because this fruit has so many virtues. An **apple** is beautiful, tasty and has a fine fragrance. It can be grown almost everywhere, and it keeps well.

On the outside it is a color that comes in various colors from green to red to gold. In taste an **apple** is tasty and juicy, and has various tastes from sour to sweet. One can eat an **apple** raw, cooked, or baked, and it can be used in various dishes, such as **apple** tzimmes, **apple** compote, **apple** pie, **apple** strudel, etc.

Apples are not planted by seeds, for, strange to say, the trees that would grow from these seeds would not necessarily remain true to the tree from which they were taken.

For instance, should the seeds of an **apple** be planted, and fruit trees result, it could happen that the **apples** would have the same taste as the **apple** from which the seeds were taken.

So, it could also happen that by a fortunate chance, an exceptionally fine kind of **apple** could result. Indeed, it was in such a manner that many of today's well-known **apples** were cultivated, and became so favored. In such cases, the area where this takes place regards it as a stroke of good fortune.

Apples are very good for one's health. An English saying is: "An **apple** a day keeps the doctor away."

Actually, an **apple** is good for one's digestion, and also has important vitamins and minerals. **Apples** also help to clean one's teeth after a meal. No other fruit has so many good qualities. Therefore, it is not surprising that **apples** are regarded as the "King" of all fruits.

The original home of the **apple** is considered to be in the countries of Southwest Asia, including the land of Israel. From there it spread to Europe and the rest of the world. **Apples** can grow anywhere, excluding extremely hot or extremely cold climates.

The trees tolerate a wide range of soil conditions, from extremely sandy soils to rather heavy clay loams; they grow especially well in intermediate types of soil. Orange orchards are generally planted in relatively deep soil where drainage is good. The orange trees are usually budded on stocks grown from the seed of selected trees. The seeds are sown in well-prepared soil in a lath house; after about 12 months' growth there, the seedlings are removed to a nursery. After about 12-16 months in the nursery, the trees are usually large enough to bud. When the budded tops are one to two years old, the trees are large enough to plant in the orchard.

TF-IDF-SIMILARITY (5/6)

- **norm** : An importance value of each field in the document is computed at indexing time and saved.
- Length of field.
Boost of field.
- Shorter fields contribute more score.

Now, why should the **apple**, in particular, be known as the "King" of all fruits? The answer is: because this fruit has so many virtues. An **apple** is beautiful, tasty and has a fine fragrance. It can be grown almost everywhere, and it keeps well.?

On the outside it is a colorful fruit. It comes in various colors from green to red to gold. In taste an **apple** is tasty and juicy, and has various tastes from sour to sweet. One can eat an **apple** raw, cooked, or baked, and it can be used in various dishes, such as **apple** tzimmes, **apple** compote, **apple** pie, **apple** strudel, etc.

Length of field = 583 characters.
Field Boost = 1 (default, but can be changed)

TF-IDF-SIMILARITY (6/6)

Lucene's Scoring Function:

$$\text{score}(q,d) = \text{coord}(q,d) \cdot \text{queryNorm}(q) \cdot \sum_{t \text{ in } q} (\text{tf}(t \text{ in } d) \cdot \text{idf}(t)^2 \cdot t.\text{getBoost}() \cdot \text{norm}(t,d))$$

Lucene's Scoring Function:

$$\text{score}(q,d) = \text{coord}(q,d) \cdot \text{queryNorm}(q) \cdot \sum_{t \in q} \left(\text{tf}(t \text{ in } d) \cdot \text{idf}(t)^2 \cdot t.\text{getBoost}() \cdot \text{norm}(t,d) \right)$$

t.getBoost - this is the search time boost given to individual terms of the query.

apple (boost 2)
orange (boost 1)

```
GET /_search
{
  "query": {
    "bool": {
      "should": [
        {
          "term": {
            "field_name": {
              "value": "apple",
              "boost": 2.0 ❶
            }
          }
        },
        {
          "term": {
            "field_name": "orange" ❷
          }
        }
      ]
    }
  }
}
```

Lucene's Scoring Function:

$$\text{score}(q,d) = \text{coord}(q,d) \cdot \text{queryNorm}(q) \cdot \sum_{t \text{ in } q} (\text{tf}(t \text{ in } d) \cdot \text{idf}(t)^2 \cdot t.\text{getBoost}() \cdot \text{norm}(t,d))$$

queryNorm - is a normalising factor to make scores between queries comparable.

The factor does not affect document ranking.

Only tries to make scores from different queries (and even different indices) comparable.

Lucene's Scoring Function:

$$\text{score}(q,d) = \text{coord}(q,d) \cdot \text{queryNorm}(q) \cdot \sum_{t \text{ in } q} (\text{tf}(t \text{ in } d) \cdot \text{idf}(t)^2 \cdot t.\text{getBoost}() \cdot \text{norm}(t,d))$$

coord - This awards additional score to documents for matching more than one query term.

ID	Query Term 1 ("Apple")	Query Term 2 ("Orange")	Hypothetical coord score
1	Yes	Yes	2
2	Yes	x	1
3	x	Yes	1
4	x	x	0

ALGORITHMS IN LUCENE

- `TFIDFSimilarity` : TF-IDF-Similarity
- **Levenshtein Algorithm**

LEVENSHTEIN DISTANCE

- Given two words, find the minimum number of operations required to change the first word into second word.
- Operations allowed:
Insert,
Delete,
Substitute

real
rearl
pearl

palace
place

drill
grill

book
bock
back

LEVENSHTEIN DISTANCE

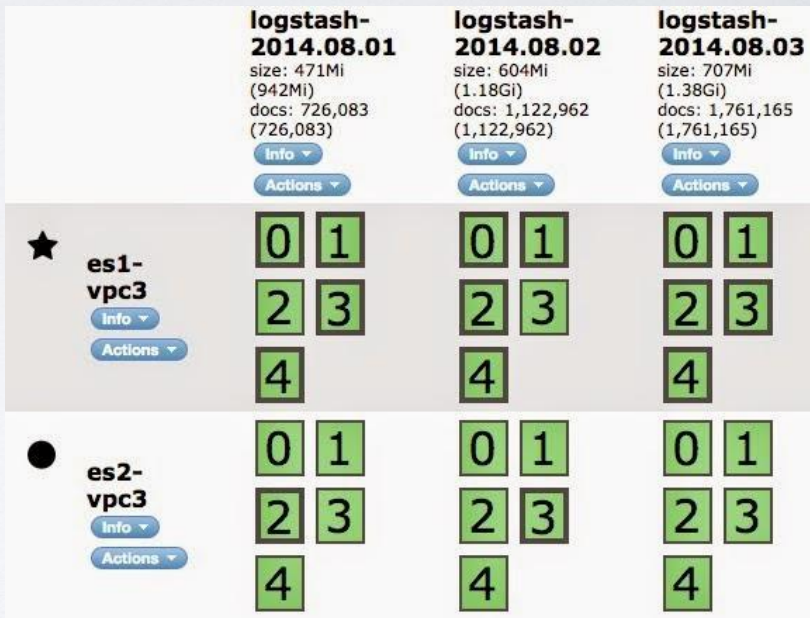
	B	O	O	K
B	0	1	2	3
A	1	1	2	3
C	2	2	2	3
K	3	3	3	2

LEVENSHTEIN DISTANCE

```
function LevenshteinDistance(char s[1..m], char t[1..n]):  
  // for all i and j, d[i,j] will hold the Levenshtein distance between  
  // the first i characters of s and the first j characters of t  
  // note that d has (m+1)*(n+1) values  
  declare int d[0..m, 0..n]  
  
  set each element in d to zero  
  
  // source prefixes can be transformed into empty string by  
  // dropping all characters  
  for i from 1 to m:  
    d[i, 0] := i  
  
  // target prefixes can be reached from empty source prefix  
  // by inserting every character  
  for j from 1 to n:  
    d[0, j] := j  
  
  for j from 1 to n:  
    for i from 1 to m:  
      if s[i] = t[j]:  
        substitutionCost := 0  
      else:  
        substitutionCost := 1  
      d[i, j] := minimum(d[i-1, j] + 1,           // deletion  
                        d[i, j-1] + 1,           // insertion  
                        d[i-1, j-1] + substitutionCost) // substitution  
  
  return d[m, n]
```


SOME ELASTICSEARCH TERMS AND IT'S ARCHITECTURE (1/3)

- An index in elastic search is similar to a DB in mysql.
- Index is divided into different shards.



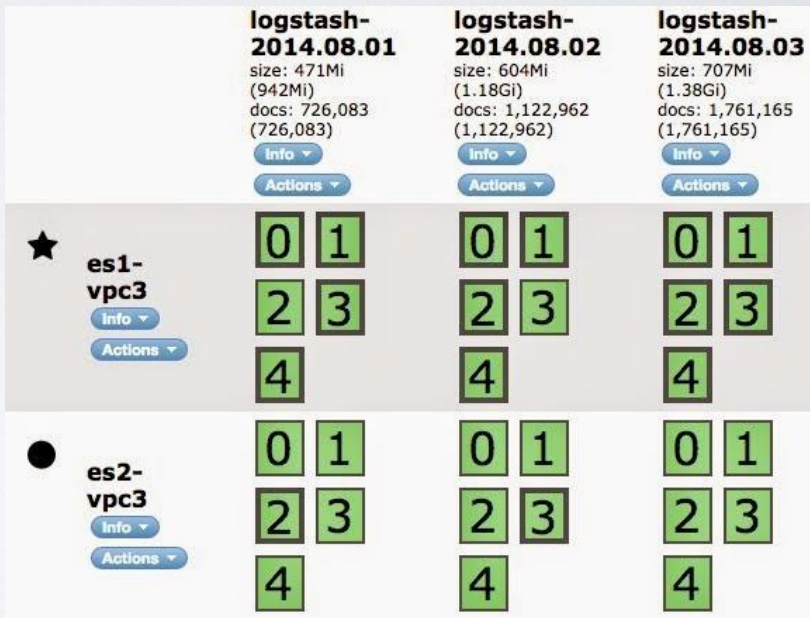
SOME ELASTICSEARCH TERMS AND IT'S ARCHITECTURE (2/3)

- A “type” in elastic search is similar to a table in mysql.
- An Index can have multiple types.

	logstash-2014.08.01 size: 471Mi (942Mi) docs: 726,083 (726,083) Info Actions	logstash-2014.08.02 size: 604Mi (1.18Gi) docs: 1,122,962 (1,122,962) Info Actions	logstash-2014.08.03 size: 707Mi (1.38Gi) docs: 1,761,165 (1,761,165) Info Actions
★ es1-vpc3 Info Actions	<div>01</div> <div>23</div> <div>4</div>	<div>01</div> <div>23</div> <div>4</div>	<div>01</div> <div>23</div> <div>4</div>
● es2-vpc3 Info Actions	<div>01</div> <div>23</div> <div>4</div>	<div>01</div> <div>23</div> <div>4</div>	<div>01</div> <div>23</div> <div>4</div>

SOME ELASTICSEARCH TERMS AND IT'S ARCHITECTURE (3/3)

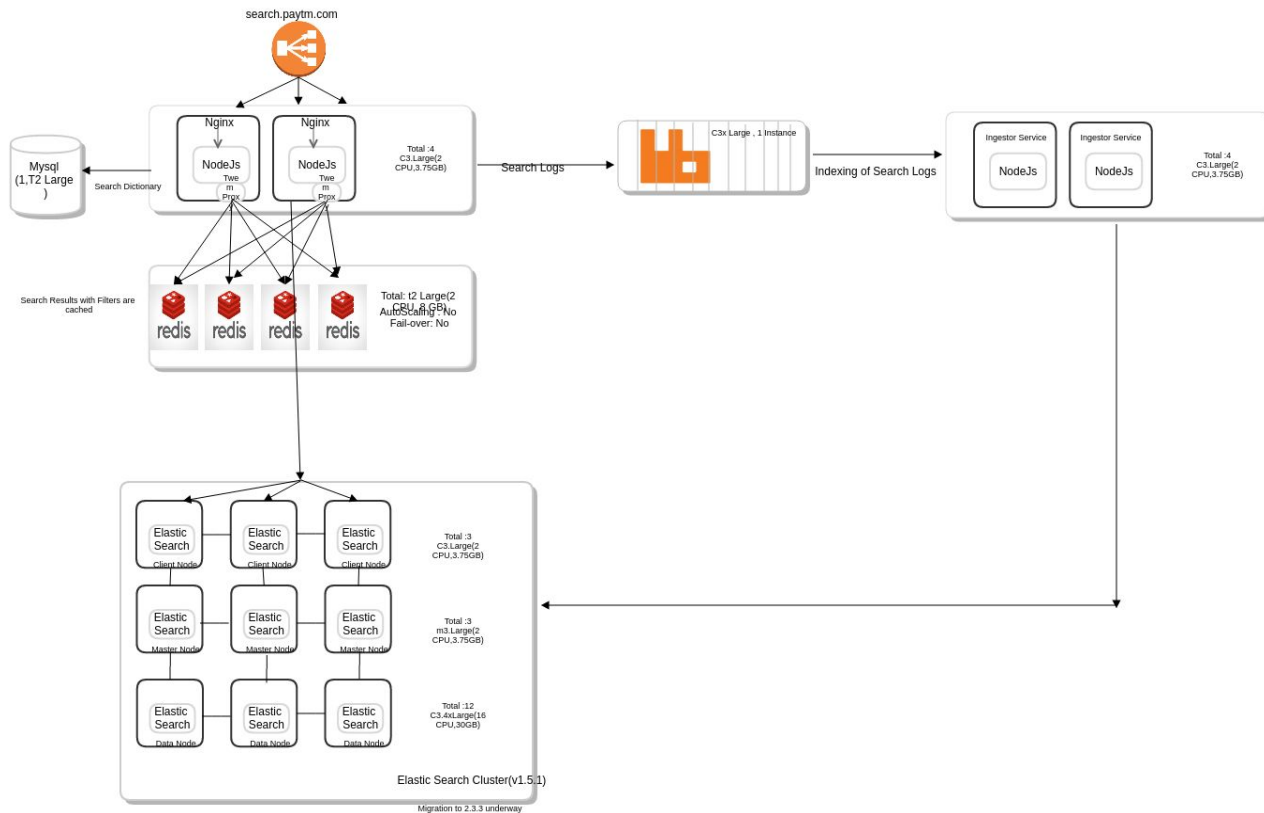
- A shard of an index contains partial data of the index that no other shard has.
- There are primary shards and replica shards. Replica and primary have exactly the same data.
- All shards are balanced across all available nodes.



ELASTICSEARCH

- We are using elastic search to power Grid, Search, Autosuggestions and Seller Panel.
- It provides very fast lookup of data.
- Elasticsearch is a distributed, RESTful search and analytics engine capable of solving a growing number of use cases. It centrally stores your data so you can discover the expected and uncover the unexpected.

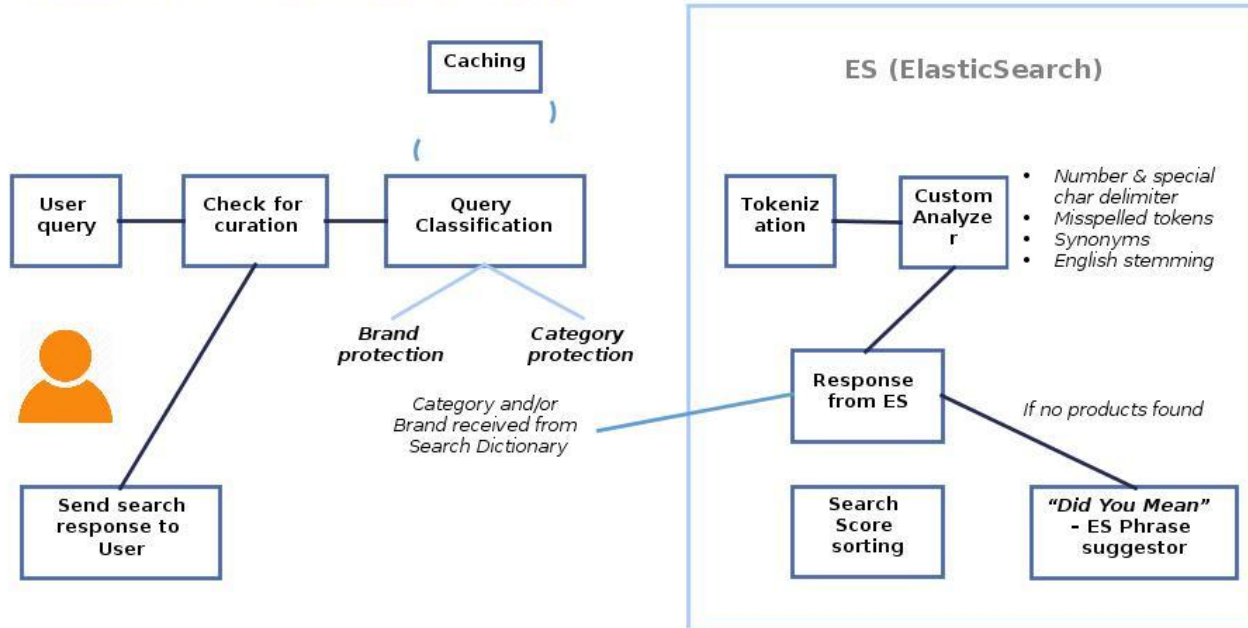
SEARCH ARCHITECTURE



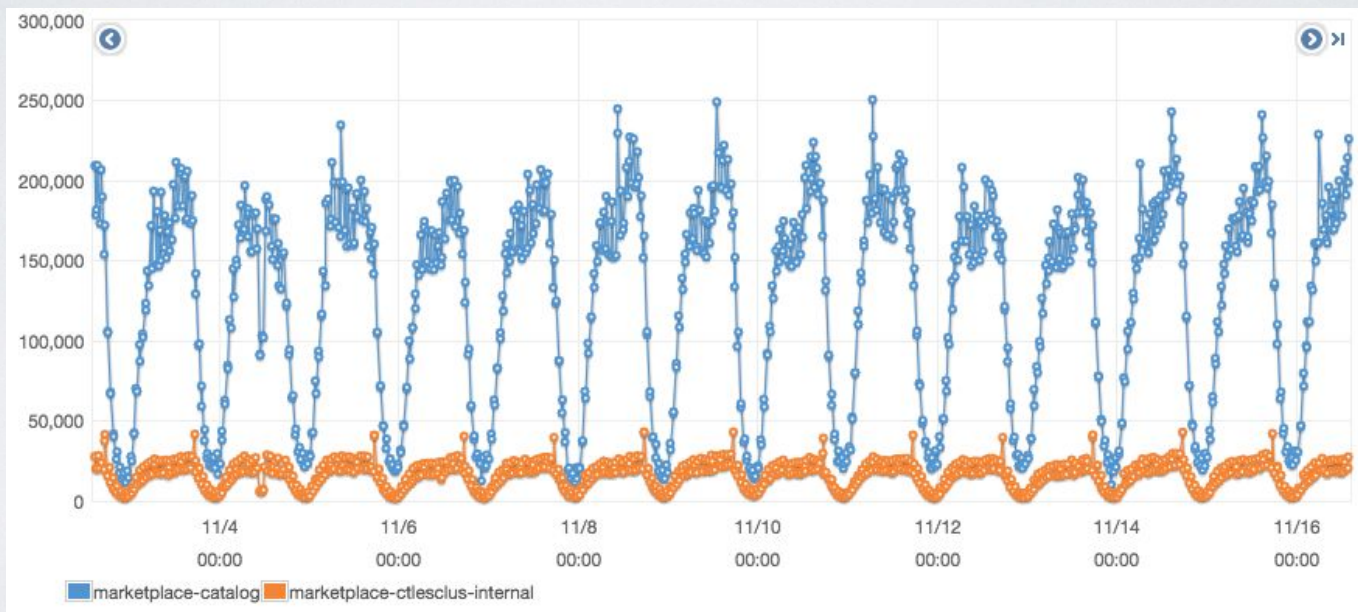
SEARCH FLOW



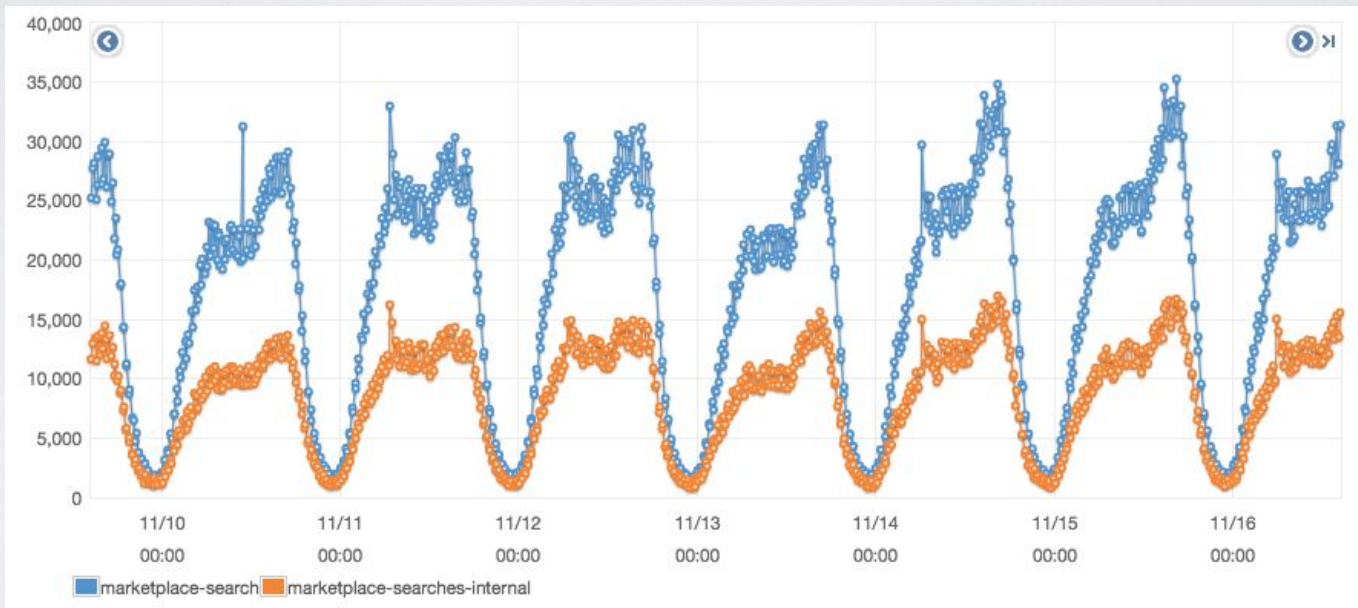
Search – Current flow



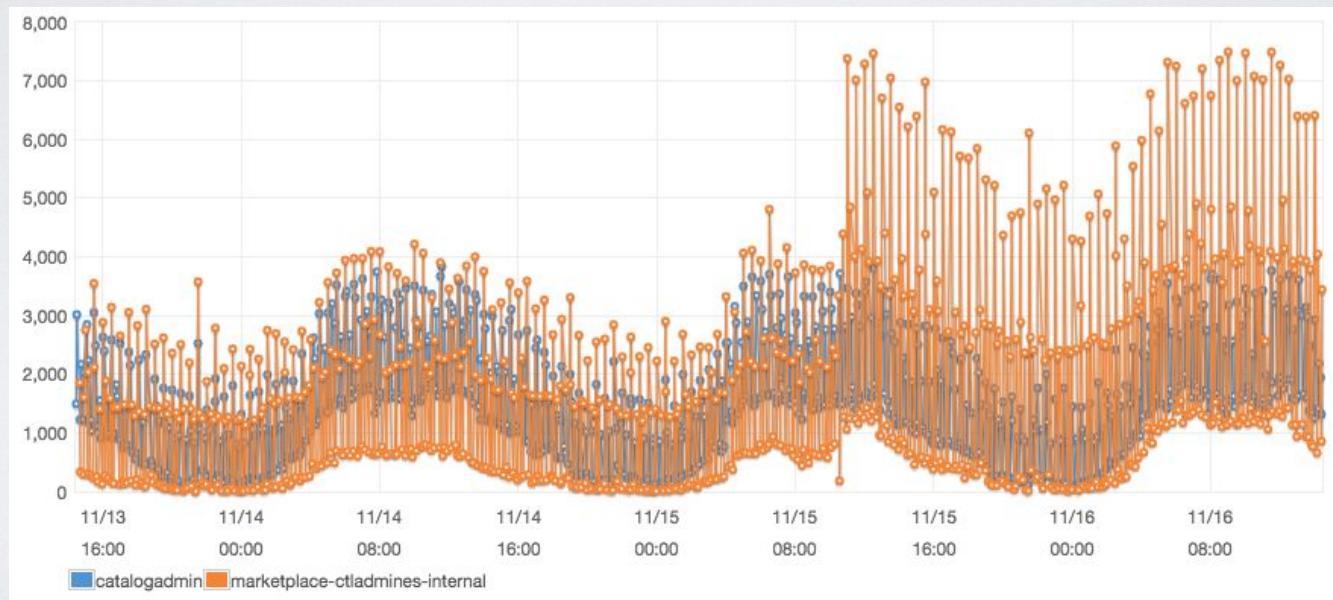
HOW MANY REQUEST ARE WE SERVING WITH ES?



HOW MANY REQUEST ARE WE SERVING WITH ES?



HOW MANY REQUEST ARE WE SERVING WITH ES?



DIFFERENCE BETWEEN SEARCH-ES AND CATALOG-ES

- Search ES contains out of stock products.
- Searches globally on the paytm database of products for matching keyword.
- Documents are scored based on brand, category, description and product name. Different weights for each.
- Catalog ES doesn't have out of stock products.
- Only serves products of a specific category ID and it's children.
- Products are displayed in a predefined order stored in ES. This order can be changed anytime using Racking flow.

DEMO ON PAYTM WEBSITE

We'll do a demo on which requests go to Search ES and which requests go to Catalog ES.

CLUSTER SETUP OF ELASTICSEARCH

- Elastic search uses zen discovery to find other nodes present in the cluster.
- We tell each node the IP of other nodes present in the cluster in a config file.
- Even if you run a single instance of elasticsearch on your local machine, you'll be running a cluster with one node.
- If any node goes down, cluster health is affected. It would turn from green to yellow or red.
- Yellow cluster state - replica shards of an index are down.
- Red cluster state - Primary shards of an index are down.

DEFINITIONS OF DIFFERENT TYPES OF NODE IN ELASTICSEARCH (1/3)

Master Node

- Responsible for creating or deleting an index.
- Tracking which nodes are part of the cluster.
- Decide which shards to allocate to which nodes.

Tip - Ensure there are dedicated master nodes to ensure stable cluster health.

DEFINITIONS OF DIFFERENT TYPES OF NODE IN ELASTICSEARCH (2/3)

Data Node

- Data nodes hold the shards.
- Handles CRUD, search and aggregations.

Tip -These operations put load on memory, IO and CPU. Monitoring them is a good idea. More data nodes should be added if these are consistently high.

DEFINITIONS OF DIFFERENT TYPES OF NODE IN ELASTICSEARCH (3/3)

Client Node

- Routing of CRUD queries.
- Basically behave as load balancers.

Tip - A large cluster can benefit from client nodes by offloading master eligible nodes and data nodes. But adding too many client nodes can slow down a cluster.

INFORMATION ON OUR CURRENT MASTER, CLIENT AND DATA NODES

ES Cluster	Master Nodes	Data Nodes	Client Nodes
Admin ES	3	4	
Catalog ES	6	16	2
Search ES	3	10	

INFORMATION ON OUR DATA SIZE, NUMBER OF DOCS AND AVG. QUERY TIME

ES Cluster	Data Size	Num Docs	Avg. Query Time
Admin ES	254.4 GB	14,59,55,294	35 ms
Catalog ES	127.83 GB	4,10,19,363	18 - 38 ms
Search ES	103 GB	2,81,60,812	17 - 22 ms

MONITORING ELASTICSEARCH



24 nodes

24 indices

331 shards

94,073,678 docs

256.64GB ↑ 15.08MB

filter indices by name

☒ closed (0)☐ special (1)

filter nodes by name

1-5 of 23 selected

	blazeds shards: 5 * 2 docs: 0 size: 795.00b	catalog shards: 5 * 2 docs: 3,082 size: 367.35KB	catalog_offline shards: 15 * 2 docs: 5,906,429 size: 9.36GB	catalog_v3 shards: 15 * 2 docs: 50,834,272 size: 95.04GB	indices catalog_v3_2017-03-20 shards: 5 * 2 docs: 0 size: 795.00b
pawslmktcatalogdatae... 10.0.44.232 heap disk cpu load	1		11 14	10 14	2
pawslmktcatalogdatae... 10.0.44.234 heap disk cpu load	4	4	4 10	2	
pawslmktcatalogdatae... 10.0.44.233 heap disk cpu load	1	2	0 6	13 14	4
pawslmktcatalogdatae... 10.0.44.11 heap disk cpu load	2	0	1	5	3
pawslmktcatalogdatae... 10.0.44.15 heap disk cpu load	3	0	3 10	6 10	0
pawslmktcatalogdatae... 10.0.44.13 heap disk cpu load			4 8	7 12	
pawslmktcatalogdatae... 10.0.44.14 heap disk cpu load	0		3 14	4 8	

https://galaxy.paytm.com/cataloges/_plugin/kopf/#!cluster

KOPF PLUGIN

CLUSTER NAME

search_es_clusterSTATUS **GREEN** NODES 13 DOCS 78 942 136

SHARDS

PRIMARY 22

RELOCATING 0

URL

INITIALIZING 0

UNASSIGNED 0

<https://galaxy.paytm.com/search>

Refresh every 5 sec Stop Sounds?

STATS

Hide



PARAMEDIC PLUGIN

NODES

pawslmktsearchdataes05

ID: ZoCc6w2ZQJSQjqUZqfHzTg
 IP: 10.0.40.134:9200
 HOST: 10.0.40.134
 LOAD: 3.200
 SIZE: 28305.57 MB
 DOCS: 17 414 138
 HEAP: 11278.58 /19213.98 MB

pawslmktsearchdataes10

ID: PGmbX8f0TIWMOSGETTo2Jw
 IP: 10.0.44.121:9200
 HOST: 10.0.44.121
 LOAD: 2.850
 SIZE: 27795.85 MB
 DOCS: 17 407 003
 HEAP: 6325.71 /19213.98 MB

pawslmktsearchdataes01

ID: -mjVQ5w4RQaCRW-g5A_DWA
 IP: 10.0.40.138:9200
 HOST: 10.0.40.138
 LOAD: 2.690
 SIZE: 27243.64 MB
 DOCS: 17 315 647
 HEAP: 5501.62 /19213.98 MB

pawslmktsearchmasteres03

ID: x4cqskamRjCR3ZNmR3-Fgw
 IP: 10.0.46.141:9200
 HOST: 10.0.46.141
 LOAD: 0.020
 SIZE: 0.00 MB
 DOCS: 0
 HEAP: 126.06 /2130.05 MB

pawslmktsearchdataes04

ID: 10N0ahLFQ6WsoOCIO307Cg
 IP: 10.0.40.135:9200
 HOST: 10.0.40.135
 LOAD: 2.870
 SIZE: 32445.69 MB
 DOCS: 17 415 763
 HEAP: 13260.99 /19213.98 MB

pawslmktsearchdataes07

ID: f4EqnOFFQoqzRIAQqWdaig
 IP: 10.0.44.123:9200
 HOST: 10.0.44.123
 LOAD: 3.230
 SIZE: 25151.01 MB
 DOCS: 17 411 214
 HEAP: 11115.90 /19213.98 MB

pawslmktsearchdataes09

ID: kRkzZy8qSCmWcmCLndut9Q
 IP: 10.0.44.119:9200
 HOST: 10.0.44.119
 LOAD: 2.270
 SIZE: 30921.88 MB
 DOCS: 17 397 745
 HEAP: 7861.20 /19213.98 MB

pawslmktsearchdataes03

ID: bEJYWI9Rc6jQsRkiAO7FA
 IP: 10.0.40.137:9200
 HOST: 10.0.40.137
 LOAD: 2.650
 SIZE: 26373.24 MB
 DOCS: 17 327 328
 HEAP: 9071.03 /19213.98 MB

pawslmktsearchdataes02

ID: 6a1IRCyfQRmtt24HOPNw_w
 IP: 10.0.40.136:9200
 HOST: 10.0.40.136
 LOAD: 3.000
 SIZE: 30426.80 MB
 DOCS: 17 415 499
 HEAP: 10725.71 /19213.98 MB

pawslmktsearchdataes06

ID: tMB7wMoWSlyxSEgw5fAseQ
 IP: 10.0.44.122:9200
 HOST: 10.0.44.122
 LOAD: 2.460
 SIZE: 28941.38 MB
 DOCS: 17 404 268
 HEAP: 8363.81 /19213.98 MB

★ pawslmktsearchmasteres02

ID: HyfX7I7_T6qGazynN_D-w
 IP: 10.0.41.63:9200
 HOST: 10.0.41.63
 LOAD: 0.070
 SIZE: 0.00 MB
 DOCS: 0
 HEAP: 668.01 /2130.05 MB

pawslmktsearchdataes08

ID: ajMv4UqjS-6eBC5WQ8rxDA
 IP: 10.0.44.120:9200
 HOST: 10.0.44.120
 LOAD: 2.410
 SIZE: 25591.55 MB
 DOCS: 17 389 993
 HEAP: 14291.91 /19213.98 MB

pawslmktsearchmasteres01

ID: u2UIAGGETJWctv5R4QK-pA
 IP: 10.0.41.62:9200
 HOST: 10.0.41.62
 LOAD: 0.070
 SIZE: 0.00 MB
 DOCS: 0
 HEAP: 246.50 /2130.05 MB

INDICES

1. autocomplete_ver2_2017_11_16_12_24

0 0 0 0 0 0 0 0 0 0

shards / replicas / 1 514 069 docs // indexing / querying / open

2. catalog_v3

0 1 2 3 4 5 6 7 8 9 0 1 2 3 4 5 6 7 8 9

shards / replicas / 63 882 377 docs // indexing / querying / open

3. dictionary_2017_11_16_18_39

0 0 0 0 0 0 0 0 0 0

shards / replicas / 487 722 docs // indexing / querying / open

PARAMEDIC PLUGIN

CLUSTER HEALTH

- GET /_cat/health?v
- Green = everything is good
- Yellow = all data is available but some replicas are missing.
- Red = Some data is not available.

ADD/REMOVE NODES

1. Fix the unicast discovery list to contain the new node.
2. Start the new node
3. Watch it join the cluster. If there's a problem, troubleshoot and repeat.
4. Use the allocation filtering API to ban all shards from the node you will be removing.
5. Wait until there are no shards on that node.
6. Shut down the node.