

What's New in Percolator?

@luccavanna

Elasticsearch User Group BeLux

Agenda

Agenda

What is the percolator?

Agenda

What is the percolator?

What do people use it for?

Agenda

What is the percolator?

What do people use it for?

How does it work?

Agenda

What is the percolator?

What do people use it for?

How does it work?

Why did it need a redesign?

Agenda

What is the percolator?

What do people use it for?

How does it work?

Why did it need a redesign?

What has changed?

Perco ...what?



We are used to docs...

We are used to docs...

```
PUT /twitter/tweet/1 -d '  
{  
  "tweet": "#elasticsearch is AWESOME",  
  "nick":  "@lucacavanna",  
  "name":  "Luca Cavanna",  
  "date":  "2013-10-02",  
  "rt":    5,  
  "loc":   {  
    "lat": 13.4,  
    "lon": 52.5  
  }  
}
```

...and queries

...and queries

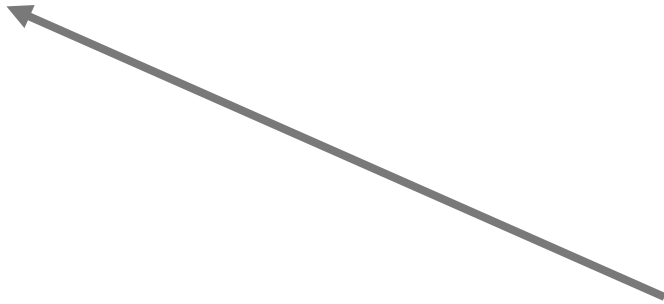
```
GET /_search -d '{
  "query": {
    "match": { "tweet": "elasticsearch" }
  }
}'
```

Let's do the **opposite**

Let's index a query...

Let's index a query...

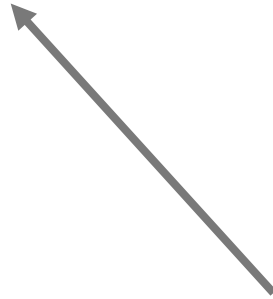
PUT `/_percolator/twitter/es-tweets`



verb (index request)

Let's index a query...

```
PUT /_percolator/twitter/es-tweets
```



reserved index

Let's index a query...

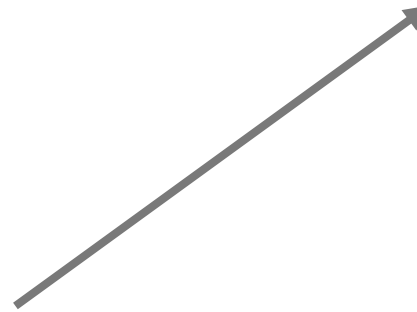
```
PUT /_percolator/twitter/es-tweets
```



index to register query against

Let's index a query...

```
PUT /_percolator/twitter/es-tweets
```



query identifier

Let's index a query...

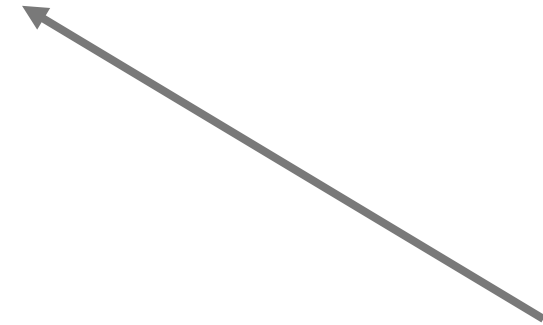
```
PUT /_percolator/twitter/es-tweets -d '  
{  
  "query": {  
    "match": { "tweet": "elasticsearch" }  
  }  
}  
,
```



the query

...and percolate a doc

GET /twitter/tweet/_percolate



verb (percolate request)

...and percolate a doc

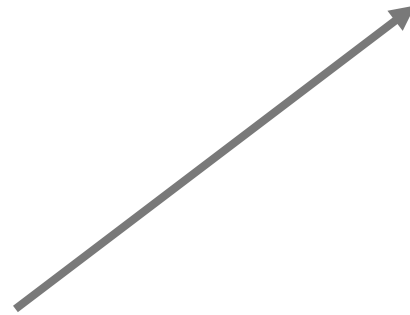
```
GET /twitter/tweet/_percolate
```

index to percolate against
(needs to exist although empty)



...and percolate a doc

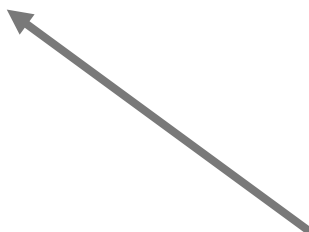
GET /twitter/tweet/_percolate



percolate api endpoint

...and percolate a doc

```
GET /twitter/tweet/_percolate -d '{
  "doc": {
    "tweet": "#elasticsearch is AWESOME",
    "nick": "@lucacavanna",
    "name": "Luca Cavanna",
    "date": "2013-10-02",
    "rt": 5,
    "loc": {
      "lat": 13.4,
      "lon": 52.5
    }
  }
}
```



document

...and percolate a doc

```
# 200 OK
```

```
{  
  "ok":    "true",  
  "matches": [ "es-tweets" ]  
}
```



matching queries

...and percolate a doc

```
# 200 OK
```

```
{  
  "ok":    "true",  
  "matches": [ "es-tweets" ]  
}
```



```
PUT /_percolator/twitter/es-tweets -d '  
{  
  "query": {  
    "match": { "tweet": "elasticsearch" }  
  }  
}'
```

Why would you do that?



Let me know **when** it's coming!

weather forecast alerts

```
PUT /_percolator/forecast/summer -d '  
{  
  "query": {  
    "range": {  
      "max" : {  
        "gte" : 30  
      }  
    }  
  }  
}
```

weather forecast alerts

```
GET /forecast/type/_percolate -d '{
  "doc" : {
    "date": "2013-07-25",
    "wind": 1,
    "min": "22",
    "max": "32",
    "rain": 0
  }
}
```

weather forecast alerts

```
# 200 OK
```

```
{  
  "ok":    "true",  
  "matches": [ "summer" ]  
}
```

weather forecast alerts

```
# 200 OK
```

```
{  
  "ok":    "true",  
  "matches": [ "summer" ]  
}
```



↑
the summer is coming!

Alerting applies also to:

Alerting applies also to:

Price monitoring

Alerting applies also to:

Price monitoring

News alerts

Alerting applies also to:

Price monitoring

News alerts

Stock alerts

Alerting applies also to:

Price monitoring

News alerts

Stock alerts

Logs monitoring

Alerting applies also to:

Price monitoring

News alerts

Stock alerts

Logs monitoring

...and many more

© Original Artist / Search ID: mbcn1703



Rights Available from CartoonStock.com

"I don't know what it means, but sales have skyrocketed since I put the 'i' in front of it."

apple iphone



apple iphone



Ad related to **apple iphone** ⓘ

Apple iPhone® vanaf €1,- - Alle iPhone® telefoons bij abo
www.gsmwijzer.be/iPhone ▼

Bestel je iPhone® vandaag nog.

Contact GSMwijzer - Nieuwste Aanbiedingen - Gratis GSM + Abonnement

How do I make my **ads** better?

Store users queries

```
PUT /_percolator/ads/user1 -d '{
  "query": {
    "match": {
      "text" : "iphone subscription",
      "operator" : "and"
    }
  }
}
```

Let's percolate the ad...

```
GET /ads/apple/_percolate -d '  
{  
  "doc" : {  
    "text": "iPhone from 1 euro",  
    "url": "http://www.myads.com/myads"  
  }  
}  
,
```

...and get feedback back

```
# 200 OK
```

```
{  
  "ok":    "true",  
  "matches": []  
}
```

No query matches




Let's modify the ad...

```
GET /ads/apple/_percolate -d '  
{  
  "doc" : {  
    "text": "iPhone subscription from 1 euro",  
    "url": "http://www.myads.com/myads"  
  }  
}  
,
```

...and get feedback back

```
# 200 OK
```

```
{  
  "ok":    "true",  
  "matches": [ "user1" ]  
}
```



**How many people would
the updated ad reach?**

Enhance documents

Store articles patterns

```
PUT /_percolator/article/apple -d '{
  "query": {
    "multi_match": {
      "fields" : ["title","content"],
      "query" : "ios apple iphone ipad"
    }
  }
}
```

percolate a new article

```
GET /article/tech/_percolate -d '{
  "doc" : {
    "title": "iOS 7 released",
    "content": "iOS 7, the seventh major
iteration of the mobile platform, includes more
than 200 new features, including a brand new
bright and colorful user interface."
  }
}
```

percolate a new article

```
GET /article/tech/_percolate -d '{
  "doc" : {
    "title": "iOS 7 released",
    "content": "iOS 7, the seventh major
iteration of the mobile platform, includes more
than 200 new features, including a brand new
bright and colorful user interface."
  }
},
{
  "ok": "true",
  "matches": [ "apple" ]
}
```

index the enriched article

```
PUT /article/tech/1 -d '{
  "title": "iOS 7 released",
  "content": "iOS 7, the seventh major iteration
of the mobile platform, includes more than 200
new features, including a brand new bright and
colorful user interface.",
  "tag" : "apple"
}
```

How does it work?

special mapping

GET `/_percolator/_mapping`

special mapping

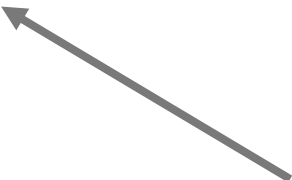
GET /_percolator/_mapping

```
{
  "_default_" : {
    "properties" : {
      "query" : {
        "type" : "object",
        "enabled" : false
      }
    }
  }
}
```

special mapping

GET /_percolator/_mapping

```
{
  "_default_" : {
    "properties" : {
      "query" : {
        "type" : "object",
        "enabled" : false
      }
    }
  }
}
```



queries are only stored

How does it work?

How does it work?

All queries are loaded in memory

How does it work?

All queries are loaded in memory

Each doc is indexed in memory

How does it work?

All queries are loaded in memory

Each doc is indexed in memory

All queries get executed against it

How does it work?

All queries are loaded in memory

Each doc is indexed in memory

All queries get executed against it

Execution time linear to # of queries

How does it work?

All queries are loaded in memory

Each doc is indexed in memory

All queries get executed against it

Execution time linear to # of queries

The memory index gets cleaned up

Query metadata

```
PUT /_percolator/twitter/es-tweets -d '  
{  
  "query": {  
    "match": { "tweet": "elasticsearch" }  
  }  
}  
,
```

Query metadata

```
PUT /_percolator/twitter/es-tweets -d '  
{  
  "query": {  
    "match": { "tweet": "elasticsearch" }  
  },  
  "user_id": "lucacavanna"  
}  
,
```


Query metadata

```
PUT /_percolator/twitter/es-tweets -d '  
{  
  "query": {  
    "match": { "tweet": "elasticsearch" }  
  },  
  "user": {  
    "name" : "Luca Cavanna",  
    "nick" : "lucacavanna"  
  }  
}  
,
```

Retrieve query by id

```
GET /_percolator/twitter/es-tweets
```

Retrieve query by id

GET `/_percolator/twitter/es-tweets`


```
{
  "query": {
    "match": { "tweet": "elasticsearch" }
  },
  "user": {
    "name" : "Luca Cavanna",
    "nick" : "lucacavanna"
  }
}
```

Drill down

```
GET /twitter/tweet/_percolate -d '{
  "doc" : {
    "tweet": "#elasticsearch is AWESOME",
    "nick": "@lucacavanna"
  },
  "query" : {
    "term" : {"user.nick" : "lucacavanna"}
  }
}
```

Drill down

```
GET /twitter/tweet/_percolate -d '{
  "doc" : {
    "tweet": "#elasticsearch is AWESOME",
    "nick": "@lucacavanna"
  },
  "query" : {
    "term" : {"user.nick" : "lucacavanna"}
  }
}
```



restricts the # of queries to run

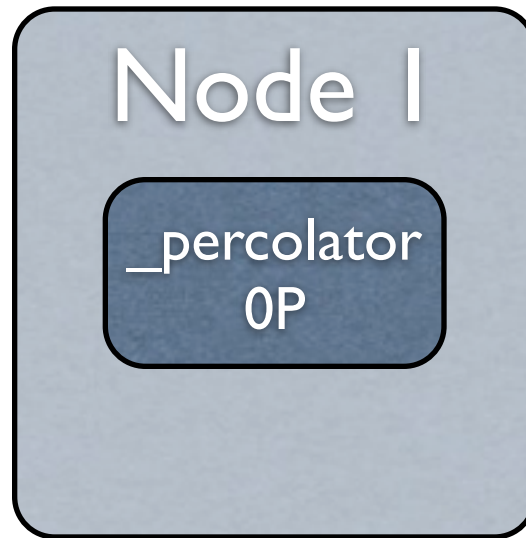
special settings

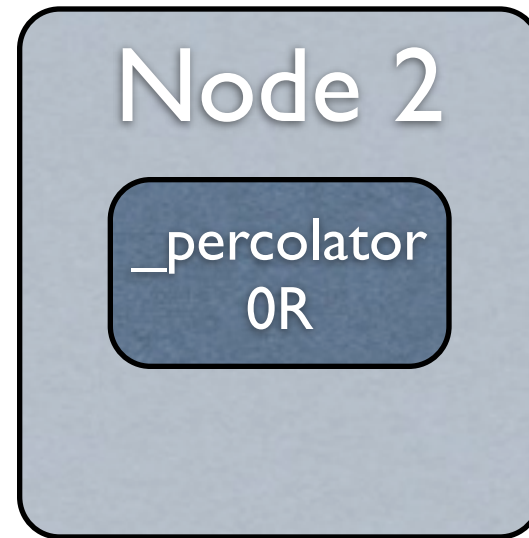
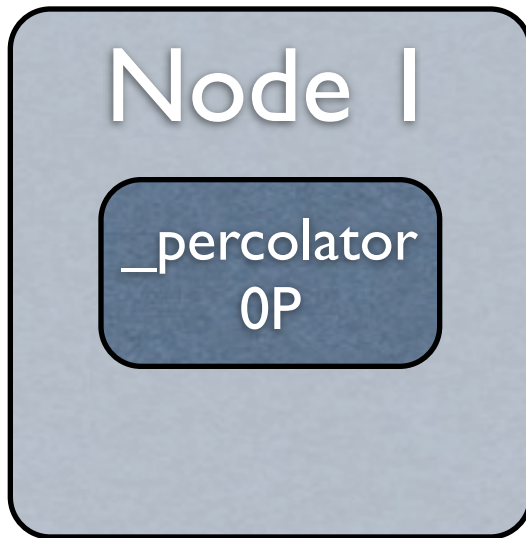
GET `/_percolator/_settings`

special settings

GET `/_percolator/_settings`

```
{
  "_percolator" : {
    "settings" : {
      "index.version.created" : "900599",
      "index.number_of_replicas" : "0",
      "index.number_of_shards" : "1",
      "index.auto_expand_replicas" : "0-all"
    }
  }
}
```





Node 1

`_percolator`
0P

Node 2

`_percolator`
0R

Node 3

`_percolator`
0R

Looks pretty **cool**,
doesn't it?

What's wrong with it
then?

A single shard doesn't
scale!


What has changed in
master (1.0)?

Let's index a query...

```
PUT /_percolator/twitter/es-tweets -d '  
{  
  "query": {  
    "match": { "tweet": "elasticsearch" }  
  }  
}  
,
```

Let's index a query...


```
PUT /anyindex/twitter/es-tweets -d '  
{  
  "query": {  
    "match": { "tweet": "elasticsearch" }  
  }  
}  
,
```



**any index can contain queries to
percolate against**

Let's index a query...

```
PUT /anyindex/_percolator/es-tweets -d '  
{  
  "query": {  
    "match": { "tweet": "elasticsearch" }  
  }  
}
```



**reserved type for queries to
percolate against**

How many **shards** now?

As **many** as you like!

This can actually **scale**!

Multi-index support

```
GET /index1,index2/tweet/_percolate -d '{
  "doc": {
    "tweet": "#elasticsearch is AWESOME",
    "nick": "@lucacavanna",
    "name": "Luca Cavanna",
    "date": "2013-10-02",
    "rt": 5,
    "loc": {
      "lat": 13.4,
      "lon": 52.5
    }
  }
},
```

Full alias support

```
GET /alias1/tweet/_percolate -d '{
  "doc": {
    "tweet": "#elasticsearch is AWESOME",
    "nick": "@lucacavanna",
    "name": "Luca Cavanna",
    "date": "2013-10-02",
    "rt": 5,
    "loc": {
      "lat": 13.4,
      "lon": 52.5
    }
  }
},
```

Routing support

```
PUT /anyindex/_percolator/id?routing=user1 -d '{
  "query": {
    "match": { "tweet": "elasticsearch" }
  }
}
```

Routing support

```
GET /twitter/tweet/_percolate?routing=user1 -d '{
  "doc": {
    "tweet": "#elasticsearch is AWESOME",
    "nick": "@lucacavanna",
    "name": "Luca Cavanna",
    "date": "2013-10-02",
    "rt": 5,
    "loc": {
      "lat": 13.4,
      "lon": 52.5
    }
  }
}
```


New percolate response

```
{
  "took": 9,
  "_shards" : {
    "total" : 5,
    "successful" : 5,
    "failed" : 0
  },
  "count" : 1,
  "matches": [
    {
      "_index" : "index1",
      "_id" : "es-tweets"
    }
  ]
}
```

New percolate response

```
{
  "took": 9,
  "_shards" : {
    "total" : 5,
    "successful" : 5,
    "failed" : 0
  },
  "count" : 1,
  "matches": [
    {
      "_index" : "index1",
      "_id" : "es-tweets"
    }
  ]
}
```

New percolate response

```
{
  "took": 9,
  "_shards" : {
    "total" : 5,
    "successful" : 5,
    "failed" : 0
  },
  "count" : 1,
  "matches": [
    {
      "_index" : "index1",
      "_id" : "es-tweets"
    }
  ]
}
```

New percolate response

```
{
  "took": 9,
  "_shards" : {
    "total" : 5,
    "successful" : 5,
    "failed" : 0
  },
  "count" : 1,
  "matches": [
    {
      "_index" : "index1",
      "_id" : "es-tweets"
    }
  ]
}
```

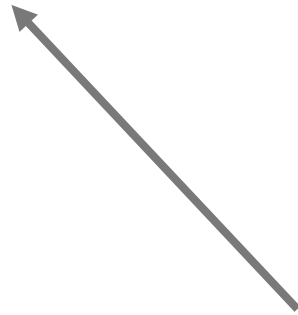
Some brand new **features**

Percolate existing doc

```
GET /twitter/tweet/1/_percolate
```

Percolate existing doc

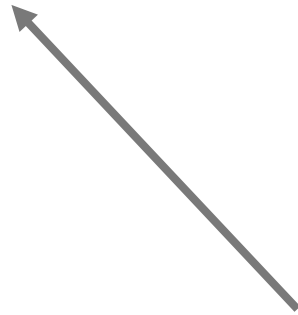
GET /**twitter**/tweet/1/_percolate



1) index that contains the document

Percolate existing doc

```
GET /twitter/tweet/1/_percolate
```



- 1) index that contains the document
- 2) and the queries (_percolator type)

Percolate existing doc

GET /twitter/**tweet**/**1**/_percolate



type & id of the existing doc

Percolate existing doc

```
GET /twitter/tweet/1/_percolate?  
percolate_index=my-percolator
```



separate index where the queries are

Percolate count api

```
GET /twitter/tweet/_percolate/count -d '{
  "doc": {
    "tweet": "#elasticsearch is AWESOME",
    "nick": "@lucacavanna",
    "name": "Luca Cavanna",
    "date": "2013-10-02",
    "rt": 5,
    "loc": {
      "lat": 13.4,
      "lon": 52.5
    }
  }
}
```

Percolate count api

```
{  
  "took": 9,  
  "_shards" : {  
    "total" : 5,  
    "successful" : 5,  
    "failed" : 0  
  },  
  "count" : 1  
}
```

Support for filters

```
GET /twitter/tweet/_percolate -d '{
  "doc" : {
    "tweet": "#elasticsearch is AWESOME",
    "nick": "@lucacavanna"
  },
  "filter" : {
    "term" : {"user.nick" : "lucacavanna"}
  }
}
```

Highlighting

```
PUT /twitter/_percolator/1 -d '  
{  
  "query": {  
    "match": { "tweet": "elasticsearch" }  
  }  
}  
,
```

Highlighting

```
PUT /twitter/_percolator/1 -d '  
{  
  "query": {  
    "match": { "tweet": "elasticsearch" }  
  }  
}
```

```
PUT /twitter/_percolator/2 -d '  
{  
  "query": {  
    "match": { "tweet": "awesome" }  
  }  
}
```

Highlighting

```
GET /twitter/tweet/_percolate -d '{
  "doc": {
    "tweet": "#elasticsearch is AWESOME"
  },
  "highlight": {
    "fields" : { "tweet":{} }
  },
  "size" : 5
}
```


Highlighting

```
{
  .....
  "matches": [
    {
      "_index" : "twitter",
      "_id" : "1",
      "tweet": "#<em>elasticsearch</em> is AWESOME"
    },
    {
      "_index" : "twitter",
      "_id" : "2",
      "tweet": "#elasticsearch is <em>AWESOME</em>"
    }
  ]
}
```

Basic scoring

```
PUT /_percolator/twitter/es-tweets -d '  
{  
  "query": {  
    "match": { "tweet": "elasticsearch" }  
  },  
  "user": {  
    "name" : "Luca Cavanna",  
    "nick" : "lucacavanna"  
  }  
}  
,
```

Basic scoring

```
PUT /_percolator/twitter/es-tweets -d '
```

```
{  
  "query": {  
    "match": { "tweet": "elasticsearch" }  
  },  
  "user": {  
    "name" : "Luca Cavanna",  
    "nick" : "lucacavanna"  
  }  
}  
,
```

```
GET /twitter/tweet/_percolate -d '
```

```
{  
  "doc" : {  
    "tweet": "#elasticsearch is AWESOME",  
    "nick": "@lucacavanna"  
  },  
  "query" : {  
    "match" : {"user.name" : "Luca"}  
  }  
}  
,
```

Basic scoring

```
PUT /_percolator/twitter/es-tweets -d '  
{  
  "query": {  
    "match": { "tweet": "elasticsearch" }  
  },  
  "user": {  
    "name" : "Luca Cavanna",  
    "nick" : "lucacavanna"  
  }  
}  
,
```

```
GET /twitter/tweet/_percolate -d '  
{  
  "doc" : {  
    "tweet": "#elasticsearch is AWESOME",  
    "nick": "@lucacavanna"  
  },  
  "query" : {  
    "match" : {"user.name" : "Luca"}  
  }  
}  
,
```



How well does
it match?

Sorting

Sort by score

Sorting

Sort by score

Only return the score

Sorting

Sort by score

Only return the score

Limit number of matches

Sorting

Sort by score

Only return the score

Limit number of matches

Boosting based on metadata

Sorting

Sort by score

Only return the score

Limit number of matches

Boosting based on metadata

Further improvements

Multi percolate

GET /_mpercolate

Multi percolate

GET /_mpercolate

```
{"percolate" : {"index":"twitter", "type":"tweet"} }  
{"doc": {"text":"elasticsearch is AWESOME!"} }  
{"percolate" : {"index":"twitter", "type":"tweet", "id":"id1"} }  
{ }  
{"count" : {"index":"twitter", "type":"tweet"} }  
{"doc": {"text":"elasticsearch rocks!"} }  
{"count" : {"index":"twitter", "type":"tweet", "id":"id2"} }  
{ }
```

www.elasticsearch.org