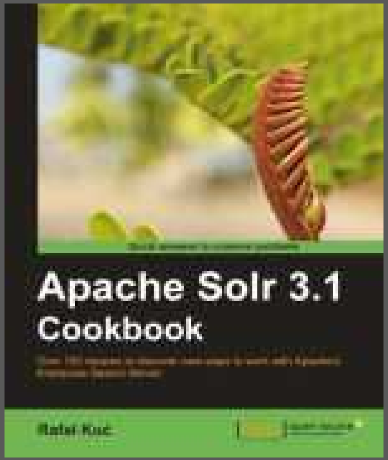
Battle of the Giants

Apache Solr 4.0 vs ElasticSearch 0.20

Rafał Kuć – Sematext International
@kucrafal @sematext sematext.com

Who Am I

- „Solr 3.1 Cookbook" author (4.0 inc)
- Sematext consultant & engineer
- Solr.pl co-founder
- Father and husband ☺



What Will I Talk About ?

elasticsearch.

VS

Apache Solr

sematext

Under the Hood

- ElasticSearch 0.20
  - Apache Lucene 3.6.1
- Apache Solr 4.0
  - Apache Lucene 4.0

Architecture

- What we expect
  - Scalability
  - Fault toleranance
  - High availablity
  - Features
- What we are also looking for
  - Manageability
  - Installation ease
  - Tools

ElasticSearch Cluster Architecture

- Distributed
- Fault tolerant
- Only ElasticSearch nodes
- Single leader
- Automatic leader election

sematext

SolrCloud Cluster Architecture

- Distributed
- Fault tolerant
- Apache Solr + ZooKeeper ensemble
- Leader per shard
- Automatic leader election

Collection vs Index

- Collection – Solr main logical index
- Index – ElasticSearch main logic structure
- Collections and Indices can be spread among different nodes in the cluster

sematext

Multiple Document Types in Index

- ElasticSearch - multiple document types in a single index
- Apache Solr - multiple document types in a single collection – shared schema.xml

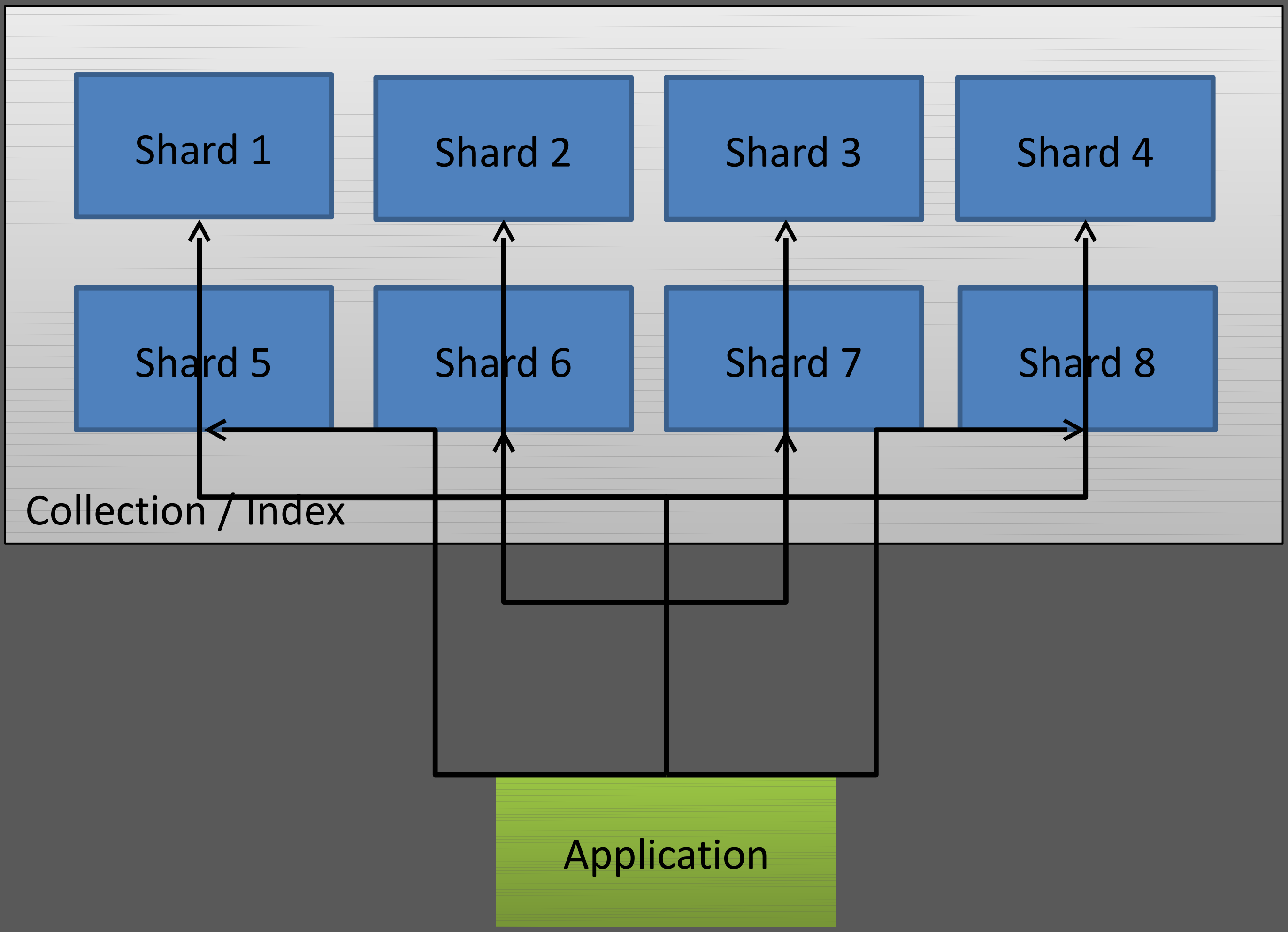
Shards and Replicas

- Index / Collection can have many shards
- Each shard can have 0 or more replicas
- Replicas are automatically updated
- Replicas can be promoted to leaders when a leader shard goes off-line

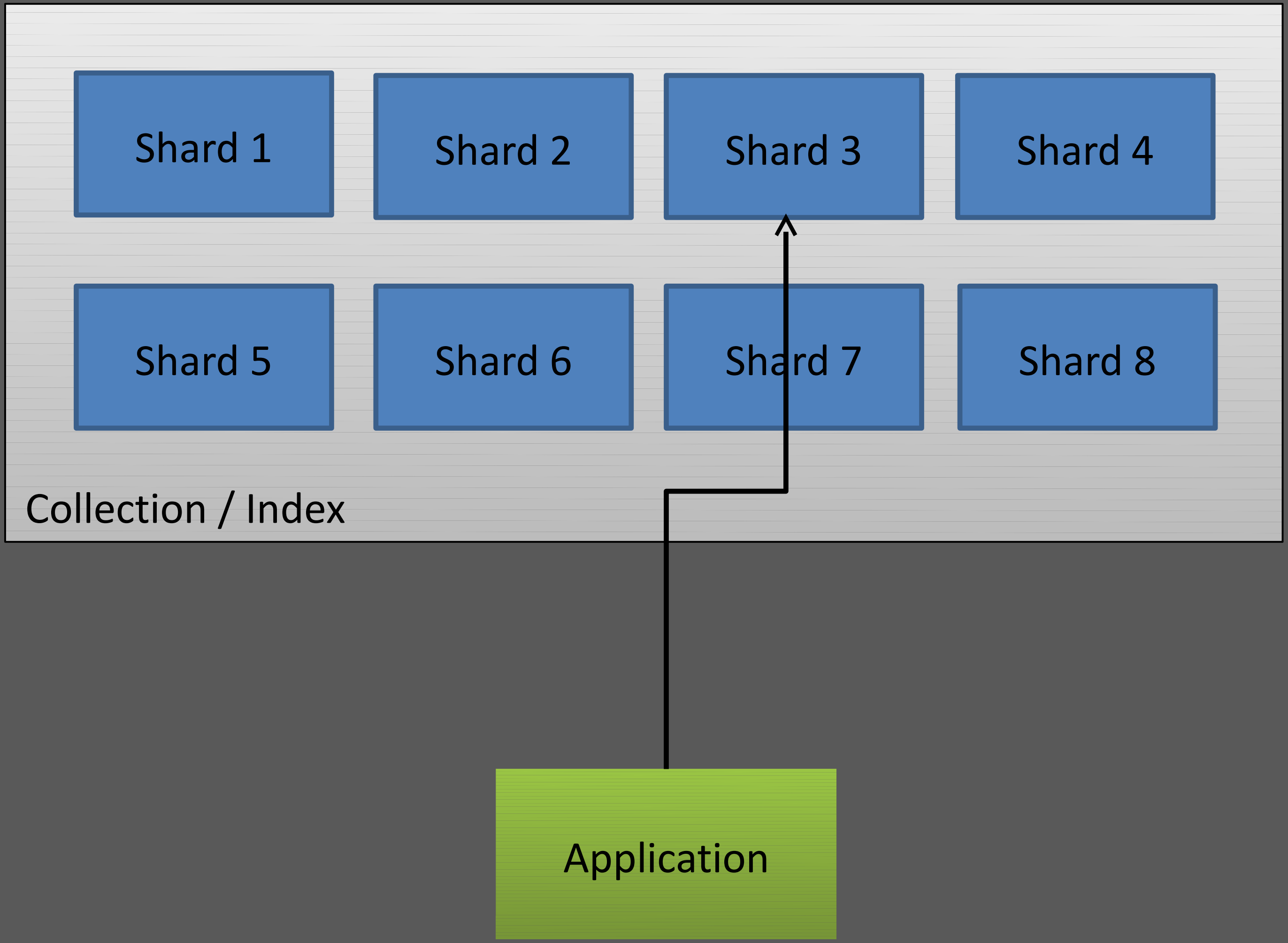
Index and Query Routing

- Control where documents are going
- Control where queries are going
- Manual data distribution

Querying Without Routing



Query With Routing



Routing Docs and Queries in Solr

- Requires some effort
- Defaults to hash based on document identifiers
- Can be turned off using `solr.NoOpDistributingUpdateProcessorFactory`

```
<updateRequestProcessorChain>
  <processor class="solr.LogUpdateProcessorFactory" />
  <processor class="solr.RunUpdateProcessorFactory" />
  <processor class="solr.NoOpDistributingUpdateProcessorFactory" />
</updateRequestProcessorChain>
```

Routing Docs and Queries - ElasticSearch

- `routing` parameter controls target shard which document/query will be forwarded to
- defaults to document identifiers
- can be changed to any value

```
curl -XPUT localhost:9200/sematext/test/1?routing=1234 -d '{
  "title" : "Test routing document"
}'

curl –XGET localhost:9200/sematext/test/_search/?q=*&routing=1234
```

Apache Solr Index Structure

- Field types defined in schema.xml file
- Fields defined in schema.xml file
- Allows automatic value copying
- Allows dynamic fields
- Allows custom similarity definition

ElasticSearch Index Structure

- Schema - less
- Analyzers and filters defined with HTTP API
- Fields defined with an HTTP request
- Multi – field support
- Allows nested documents
- Allows parent – child relationship
- Allows structured data

Index Structure Manipulation

- Possible to some extent in Solr as well as ElasticSearch
- ElasticSearch allows dynamic mappings update (not always)

Aliasing

- Solr
  - Allows core aliasing
- ElasticSearch
  - Allows index aliasing
  - We can add filter to alias
  - We can add index routing
  - We can add search routing

Server Configuration

- Solr
  - Static in `solrconfig.xml`
  - Can be reloaded during runtime with collection/core reload

- ElasticSearch
  - Static in `elasticsearch.yml`
  - Properties can be changed during runtime (although not all) without reloading

ElasticSearch Gateway Module

- Your data time machine
- Stores indices and meta data
- Currently available:
  - Local
  - Shared FS
  - Hadoop
  - S3

Discovery

- Apache Solr uses ZooKeeper
- ElasticSearch uses Zen Discovery

sematext

ElasticSearch Zen Discovery

- Allows automatic node discovery
- Provides multicast and unicast discovery methods
- Automatic master detection
- Two - way failure detection

sematext

Apache Solr & Apache ZooKeeper

- Requires additional software
- ZooKeeper ensemble with 1+ ZooKeeper instances
- Prevents split – brain situations
- Holds collections configurations
- Solr needs to know address of one of the ZooKeeper instances

API

- HTTP REST API in ElasticSearch or Query String for simple queries
- HTTP with Query String in Apache Solr
- Both provide specialized Java API
  - SolrJ for Apache Solr and CloudSolrServer
  - ElasticSearch with TransportClient for remote connections

Apache Solr and Query String

- Queries are built of request parameters
- Some degree of structuring allowed (local params)

```
curl 'http://localhost:8983/solr/select?q=text:weird&sort=date+desc'
```

ElasticSearch REST End-Points

- Simple queries built of request parameters
- Stuctured queries built as JSON objects

```
curl –XGET 'localhost:9200/sematext/test/_search/?
q=_all:weird&sort=date:desc'

curl -XGET 'localhost:9200/sematext/test_search' -d '{
"query" : {
  "term" : {
    "_all" : "weird"
  },
  "sort" : {
    "date" : {
      "order" : "desc"
    }
  }
}'
```

sematext

Data Handling

- Solr
  - Multiple formats allowed as input
  - Can return results in multiple formats
- ElasticSearch
  - JSON in / JSON out

Single or Batch

- Solr
  - Single or multiple documents per request

- ElasticSearch
  - Single document with a standard indexing call
  - `_bulk` end – point exposed for batch indexing
  - `_bulk` UDP end – point can be exposed for low latency batch indexing

Partial Document Updates

- Not based on LUCENE-3837 proposed by Andrzej Białecki
- Document reindexing on the side of search server
- Both servers use versioning to prevent changes being overwritten
- Can lead to decreased network traffic in some cases

sematext

ElasticSearch Partial Doc Update

- Special end – point exposed - `_update`
- Supports parameters like `routing`, `parent`, `replication`, `percolate`, etc (similar to Index API)
- Uses scripts to perform document updates

```
curl -XPOST 'localhost:9200/sematext/test/12345/_update' -d '{
  "script" : "ctx._source.enabled = enabled",
  "params" : {
    "enabled" : true
  }
}'
```

Apache Solr Partial Doc Update

- Sent to the standard update handler
- Requires `_version_` field to be present

```
curl 'localhost:8983/solr/update?commit=true' -H 'Content-
type:application/json' -d '[
 {
   "id" : "12345",
   "enabled" : {
     "set" : true
   }
 }
]'
```

Solr Collections API

- Built on top of Core Admin
- Allows:
  - Collection creation
  - Collection reload
  - Collection deletion

ElasticSearch Indices REST API

- Allows:
  - Index creation
  - Index deletion
  - Index closing and opening
  - Index refreshing
  - Existence checking

Analysis Chain Definition

- Solr
  - Static in `schema.xml`
  - Can be reloaded during runtime with collection/core reload

- ElasticSearch
  - Static in `elasticsearch.yml`
  - Defined during index/type creation with REST call
  - Possible to change with update mapping call (not all changes allowed)

Multilingual Data Handling

- Both ElasticSearch and Apache Solr built on top of Apache Lucene
- Solr – analyzers defined per field in `schema.xml` file
- ElasticSearch – analyzer defined in mappings, but can be set during query or specified on the basis of field values

sematext

Results Grouping

- Available in Apache Solr only
- Allows for results grouping based on:
  - Field value
  - Query
  - Function query (not available during distributed searching)

Prospective Search

- Allows for checking if a document matches a stored query
- Not available in Apache Solr
- Available in ElasticSearch under the name of Percolator

Spellchecker

- Allows to check and correct spelling mistakes
- Not available in ElasticSearch currently
- Multiple implementations available in Apache Solr
  - IndexBasedSpellChecker
  - WordBreakSolrSpellChecker
  - DirectSolrSpellChecker

Full Text Search Capabilities

- Variety of queries
- Ability to control score calculation
- Different query parsers available
- Advanced Lucene queries (like SpanQueries) exposed

sematext

Score Calculation

- Leverage Lucene scoring capabilities
- Control over document importance
- Control over query importance
- Control over term and phrase importance

Apache Solr and Score Influence

- Index time
  - Document boosts
  - Field boosts
- Query time
  - Term boosts
  - Field boosts
  - Phrases boost
  - Function queries

ElasticSearch and Score Influence

- Index time
  - Document and field boosts
- Query time
  - Different queries provide different boost controls
  - Can calculate distributed term frequencies
  - Negative and Positive boosting queries
  - Custom score filters
- Scripts

Nested Objects

- Possible only in ElasticSearch
- Indexed as separate documents
- Stored in the same part of the index as the root document
- Hidden from standard queries and filters
- Need appropriate queries and filters (nested)

More Like This

- Lets us find similar documents
- Solr
  - More Like This Component
- ElasticSearch
  - More Like This Query
  - More Like This Field Query
  - `_mlt` REST end – point

Solr Parent – Child Relationship

- Used at query time
- Multi core joins possible

[redacted]

ElasticSearch Parent – Child Handling

- Proper indexing required
- Indexed as separate documents
- Standard queries don’t return child documents
- In order to retrieve parent docs one should use appropriate queries and filters (`has_child`, `has_parent`, `top_children`)

Filters

- Used to narrown down query results
- Good candidates for caching and reuse
- Supported by ElasticSearch and Apache Solr
- Should be used for repeatable query elements

Apache Solr Filter Queries

- Multiple filters per query
- Filters are addictive
- Different query parsers can be used
- Local params can be used
- Narrow down faceting results

ElasticSearch Filtered Queries

- Can be defined using queries exposed by the Query DSL
- Can be used for custom score calculation (i.e., `custom filters score query`)
- Doesn’t narrow down faceting results by default (facets have their own filters)

Filter Cache Control

- Both Solr and ElasticSearch let us control cache for filters
- Solr
  - Using local params and cache property
- ElasticSearch
  - _cache property
  - _cache_key property

Faceting

- Both provide common facets
  - Terms
  - Range & query
  - Terms statistics
  - Spatial distance
- Solr
  - Pivot faceting
- ElasticSearch

Real Time Or Not ?

- Allow getting document not yet indexed
- Don’t need searcher reopening
- ElasticSearch
  - Separate Get and Multi Get API’s
- Apache Solr
  - Separate Realtime Get Handler
  - Can be used as a search component

Caches and Warming

- ElasticSearch and Solr allow caching
- Both allow running warming queries
- ElasticSearch by default doesn’t limit cache sizes

Solr Caches

- Types
  - Filter Cache
  - Query Result Cache
  - Document Cache
- Implementation choices
  - LRUCache
  - FastLRUCache
  - LFUCache
- Other confi options:

ElasticSearch Caches

- Types
  - Filter Cache
  - Field Data Cache
- Implementation choices
  - Resident
  - Soft
  - Weak
- Other configuration options:

Cluster State Monitoring

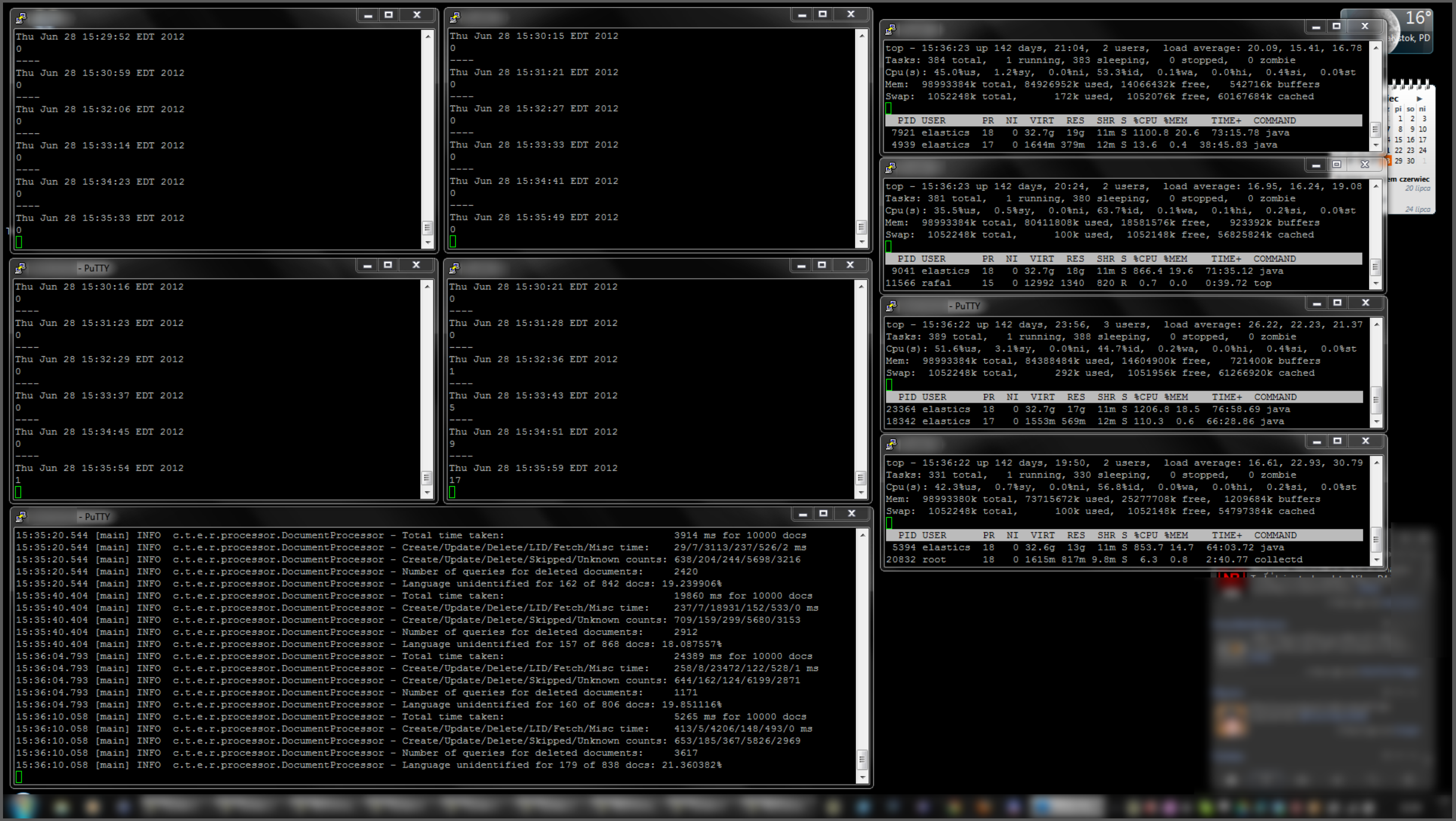
- Apache Solr – multiple mbeans exposed by JMX
- ElasticSearch – multiple REST end – points exposed to get different statistics

ElasticSearch Statistics API

- Health and State Check
- Nodes Information and Statistics
- Cache Statistics
- Index Segments Information
- Index Information and Statistics
- Mappings Information

sematext

Cluster Monitoring



```
Thu Jun 28 15:29:52 EDT 2012
0
----
Thu Jun 28 15:30:59 EDT 2012
0
----
Thu Jun 28 15:32:06 EDT 2012
0
----
Thu Jun 28 15:33:14 EDT 2012
0
----
Thu Jun 28 15:34:23 EDT 2012
0
----
Thu Jun 28 15:35:33 EDT 2012
0
```

```
Thu Jun 28 15:30:15 EDT 2012
0
----
Thu Jun 28 15:31:21 EDT 2012
0
----
Thu Jun 28 15:32:27 EDT 2012
0
----
Thu Jun 28 15:33:33 EDT 2012
0
----
Thu Jun 28 15:34:41 EDT 2012
0
----
Thu Jun 28 15:35:49 EDT 2012
0
```

```
Thu Jun 28 15:30:16 EDT 2012
0
----
Thu Jun 28 15:31:23 EDT 2012
0
----
Thu Jun 28 15:32:29 EDT 2012
0
----
Thu Jun 28 15:33:37 EDT 2012
0
----
Thu Jun 28 15:34:45 EDT 2012
0
----
Thu Jun 28 15:35:54 EDT 2012
1
```

```
Thu Jun 28 15:30:21 EDT 2012
0
----
Thu Jun 28 15:31:28 EDT 2012
0
----
Thu Jun 28 15:32:36 EDT 2012
1
----
Thu Jun 28 15:33:43 EDT 2012
5
----
Thu Jun 28 15:34:51 EDT 2012
9
----
Thu Jun 28 15:35:59 EDT 2012
17
```

```
15:35:20.544 [main] INFO  c.t.e.r.processor.DocumentProcessor - Total time taken:                         3914 ms for 10000 docs
15:35:20.544 [main] INFO  c.t.e.r.processor.DocumentProcessor - Create/Update/Delete/LID/Fetch/Misc time:    29/7/3113/237/526/2 ms
15:35:20.544 [main] INFO  c.t.e.r.processor.DocumentProcessor - Create/Update/Delete/Skipped/Unknown counts: 638/204/244/5698/3216
15:35:20.544 [main] INFO  c.t.e.r.processor.DocumentProcessor - Number of queries for deleted documents:     2420
15:35:20.544 [main] INFO  c.t.e.r.processor.DocumentProcessor - Language unidentified for 162 of 842 docs: 19.239906%
15:35:40.404 [main] INFO  c.t.e.r.processor.DocumentProcessor - Total time taken:                         19860 ms for 10000 docs
15:35:40.404 [main] INFO  c.t.e.r.processor.DocumentProcessor - Create/Update/Delete/LID/Fetch/Misc time:    237/7/18931/152/533/0 ms
15:35:40.404 [main] INFO  c.t.e.r.processor.DocumentProcessor - Create/Update/Delete/Skipped/Unknown counts: 709/159/299/5680/3153
15:35:40.404 [main] INFO  c.t.e.r.processor.DocumentProcessor - Number of queries for deleted documents:     2912
15:35:40.404 [main] INFO  c.t.e.r.processor.DocumentProcessor - Language unidentified for 157 of 868 docs: 18.087557%
15:36:04.793 [main] INFO  c.t.e.r.processor.DocumentProcessor - Total time taken:                         24389 ms for 10000 docs
15:36:04.793 [main] INFO  c.t.e.r.processor.DocumentProcessor - Create/Update/Delete/LID/Fetch/Misc time:    258/8/23472/122/528/1 ms
15:36:04.793 [main] INFO  c.t.e.r.processor.DocumentProcessor - Create/Update/Delete/Skipped/Unknown counts: 644/162/124/6199/2871
15:36:04.793 [main] INFO  c.t.e.r.processor.DocumentProcessor - Number of queries for deleted documents:     1171
15:36:04.793 [main] INFO  c.t.e.r.processor.DocumentProcessor - Language unidentified for 160 of 806 docs: 19.851116%
15:36:10.058 [main] INFO  c.t.e.r.processor.DocumentProcessor - Total time taken:                         5265 ms for 10000 docs
15:36:10.058 [main] INFO  c.t.e.r.processor.DocumentProcessor - Create/Update/Delete/LID/Fetch/Misc time:    413/5/4206/148/493/0 ms
15:36:10.058 [main] INFO  c.t.e.r.processor.DocumentProcessor - Create/Update/Delete/Skipped/Unknown counts: 653/185/367/5826/2969
15:36:10.058 [main] INFO  c.t.e.r.processor.DocumentProcessor - Number of queries for deleted documents:     3617
15:36:10.058 [main] INFO  c.t.e.r.processor.DocumentProcessor - Language unidentified for 179 of 838 docs: 21.360382%
```

```
top - 15:36:23 up 142 days, 21:04,  2 users,  load average: 20.09, 15.41, 16.78
Tasks: 384 total,   1 running, 383 sleeping,   0 stopped,   0 zombie
Cpu(s): 45.0%us,  1.2%sy,  0.0%ni, 53.3%id,  0.1%wa,  0.0%hi,  0.4%si,  0.0%st
Mem:  98993384k total, 84926952k used, 14066432k free,   542716k buffers
Swap:  1052248k total,      172k used,  1052076k free, 60167684k cached

  PID USER      PR  NI  VIRT  RES  SHR S %CPU %MEM    TIME+  COMMAND
 7921 elastics  18   0 32.7g  19g  11m S 1100.8 20.6  73:15.78 java
 4939 elastics  17   0 1644m 379m  12m S 13.6  0.4  38:45.83 java
```

```
top - 15:36:23 up 142 days, 20:24,  2 users,  load average: 16.95, 16.24, 19.08
Tasks: 381 total,   1 running, 380 sleeping,   0 stopped,   0 zombie
Cpu(s): 35.5%us,  0.5%sy,  0.0%ni, 63.7%id,  0.1%wa,  0.1%hi,  0.2%si,  0.0%st
Mem:  98993384k total, 80411808k used, 18581576k free,   923392k buffers
Swap:  1052248k total,      100k used,  1052148k free, 56825824k cached

  PID USER      PR  NI  VIRT  RES  SHR S %CPU %MEM    TIME+  COMMAND
 9041 elastics  18   0 32.7g  18g  11m S 866.4 19.6  71:35.12 java
11566 rafal     15   0 12992 1340  820 R  0.7  0.0   0:39.72 top
```

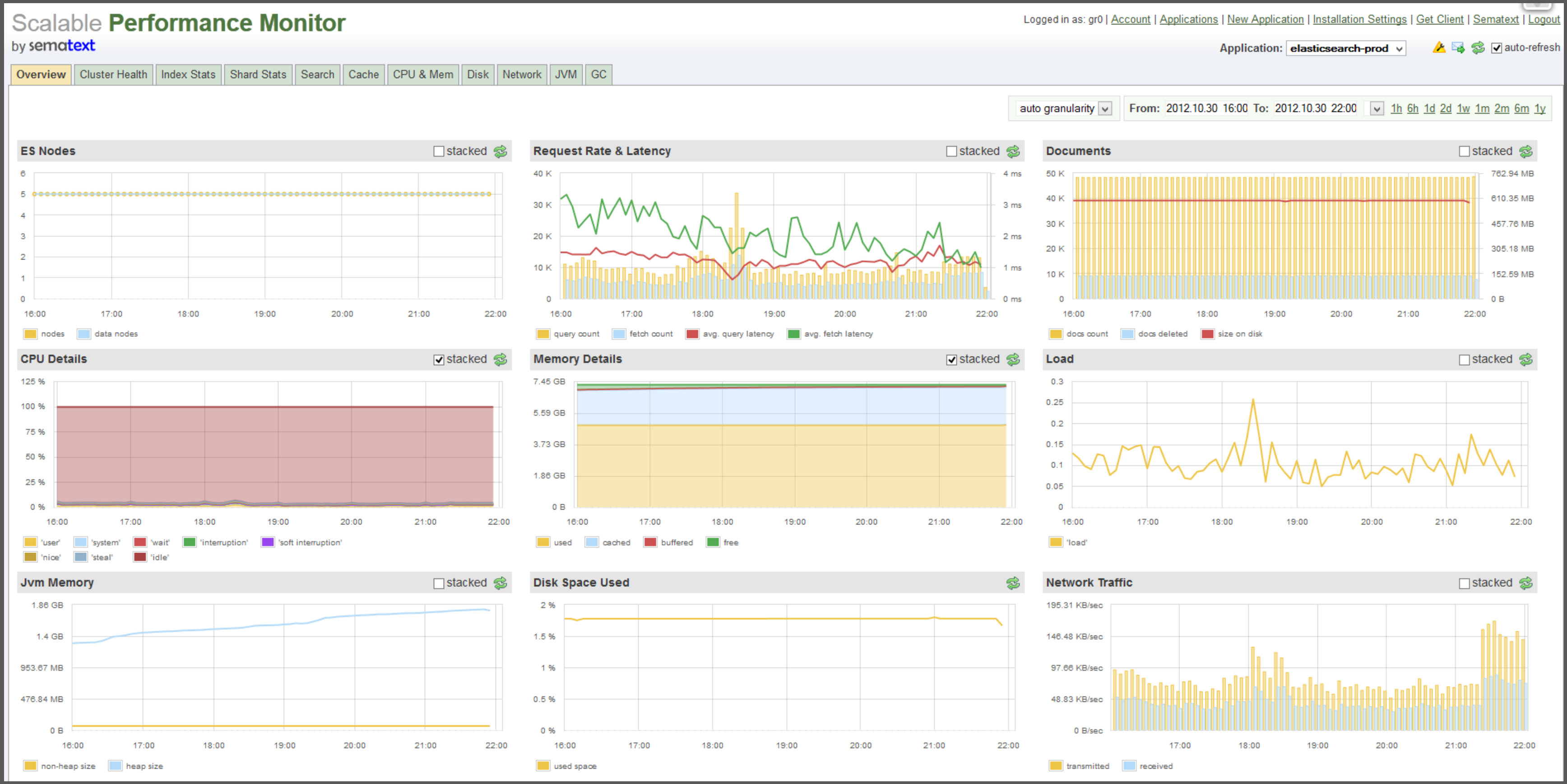
```
top - 15:36:22 up 142 days, 23:56,  3 users,  load average: 26.22, 22.23, 21.37
Tasks: 389 total,   1 running, 388 sleeping,   0 stopped,   0 zombie
Cpu(s): 51.6%us,  3.1%sy,  0.0%ni, 44.7%id,  0.2%wa,  0.0%hi,  0.4%si,  0.0%st
Mem:  98993384k total, 84388484k used, 14604900k free,   721400k buffers
Swap:  1052248k total,      292k used,  1051956k free, 61266920k cached

  PID USER      PR  NI  VIRT  RES  SHR S %CPU %MEM    TIME+  COMMAND
23364 elastics  18   0 32.7g  17g  11m S 1206.8 18.5  76:58.69 java
18342 elastics  17   0 1553m 569m  12m S 110.3  0.6  66:28.86 java
```

```
top - 15:36:22 up 142 days, 19:50,  2 users,  load average: 16.61, 22.93, 30.79
Tasks: 331 total,   1 running, 330 sleeping,   0 stopped,   0 zombie
Cpu(s): 42.3%us,  0.7%sy,  0.0%ni, 56.8%id,  0.0%wa,  0.0%hi,  0.2%si,  0.0%st
Mem:  98993380k total, 73715672k used, 25277708k free,  1209684k buffers
Swap:  1052248k total,      100k used,  1052148k free, 54797384k cached

  PID USER      PR  NI  VIRT  RES  SHR S %CPU %MEM    TIME+  COMMAND
 5394 elastics  18   0 32.6g  13g  11m S 853.7 14.7  64:03.72 java
20832 root      18   0 1615m 817m 9.8m S  6.3  0.8   2:40.77 collectd
```

Cluster Monitoring with SPM


Logged in as: gr0 | Account | Applications | New Application | Installation Settings | Get Client | Sematext | Logout

Application: elasticsearch-prod ✓ auto-refresh

auto granularity From: 2012.10.30 16:00 To: 2012.10.30 22:00 1h 6h 1d 2d 1w 1m 2m 6m 1y

sematext

Cluster Settings Update

- ElasticSearch lets us:
  - Control rebalancing
  - Control recovery
  - Control allocation
  - Change the above on the live cluster

sematext

Custom Shard Allocation

- Possible in ElasticSearch
- Cluster level:

```
curl -XPUT localhost:9200/_cluster/settings -d '{
   "persistent" : {
      "cluster.routing.allocation.exclude._ip" : "192.168.2.1"
   }
}'
```

```
curl -XPUT localhost:9200/sematext/ -d '{
   "index.routing.allocation.include.tag" : "nodeOne,nodeTwo"
}'
```

Moving Shards and Replicas

- Possible in ElasticSearch, not available in Solr
- Allows to move shards and replicas to any node in the cluster on demand
- Available in ElasticSearch:

[redacted]

sematext

And The Winner Is ?

The Users

How to Reach Us

- Rafał Kuć
  - Twitter: @kucrafal
  - E-mail: rafal.kuc@sematext.com
- Sematext
  - Twitter: @sematext
  - Website: http://sematext.com
- Solr vs ElasticSearch series:
  - http://blog.sematext.com/2012/08/23/solr-vs-elasticsearch-par

sematext

We Are Hiring !

- Dig Search ?
- Dig Analytics ?
- Dig Big Data ?
- Dig Performance ?
- Dig working with and in open – source ?
- We're hiring world – wide !

http://sematext.com/about/jobs.html

sematext