ELASTICSEARCH

ELASTICSEARCH – HOW TO SETUP, TOOLS, NICE STUFF AND PITFALLS

AGENDA

/ME

- Lars Erler
- Team Lead @moebel
- 15 years work experience
  - Team Lead @Notebooksbilliger,
  - Team Lead @Lamudi (Rocket Venture),
  - CTO @Bringmeister (Kaiser's Tengelmann)
- Passioned about automation, open source, delivering good software for the customer and fixing problems @root cause

ELASTIC STACK

- Initial release 2010
- Current release 6.6
- Open Source - Apache2 License (except commercial part x-pack)
- Elasticsearch, Kibana, Beats and Logstash (before own projects)
- "Elasticsearch is a search engine based on the Lucene library. It provides a distributed, multitenant-capable full-text search engine with an HTTP web interface and schema-free JSON documents"*
- "Elasticsearch uses Lucene and tries to make all its features available through the JSON and Java API. It supports facetting and percolating"*

* https://en.wikipedia.org/wiki/Elasticsearch

LUCENE

- Apache Lucene is a free and open-source information retrieval software library
- Initial Release 1999, actual Version 7.6.0 (December 2018)
- "While suitable for any application that requires full text indexing and searching capability, Lucene has been widely recognized for its utility in the implementation of Internet search engines and local, single-site searching“
- "Lucene itself is just an indexing and search library“

* https://en.wikipedia.org/wiki/Apache_Lucene

LUCENE

- Features
    - ranked searching -- best results returned first
    - many powerful query types: phrase queries, wildcard queries, proximity queries, range queries and more
    - fielded searching (e.g. title, author, contents)
    - sorting by any field
    - multiple-index searching with merged results
    - allows simultaneous update and searching
    - flexible faceting, highlighting, joins and result grouping
    - fast, memory-efficient and typo-tolerant suggesters

* http://lucene.apache.org/

ELASTICSEARCH VS. SOLR

- Working with Solr since 2009, Elasticsearch since 2017
- ES was built with REST as primary interface
- Solr is best suited with Java Clients
- Elasticsearch's Query DSL syntax is really flexible and it's pretty easy to write complex queries with it, compared to DisMax Parser in Solr
- Both are mature and stable, Features are very similar
- Ecosystem around Elasticsearch it is very modern and sorted
- ES is backed by commercial entity, which is worth 6 billion $$ at the moment, Solr is Apache Project
- Elasticsearch provides a powerful aggregations engine that not only can do one level data analysis like most of the Solr legacy facets, but can also nest data analysis (e.g., calculate average price for each product category in each shop division)
- Additional reads
  - http://solr-vs-elasticsearch.com/
  - https://sematext.com/blog/solr-vs-elasticsearch-differences/

USAGE @MOEBEL

- We use ES primarily for our Frontend as Live Storage, our DB is only available in the backend
    - Sync of data via partial and full update workers
- Used Indices Frontend
    - Product
    - Navigation
    - CMS
    - Seller
    - Equipment Percolate & Equipment Rules (Talk next time from Kevin)
    - Autocompletion
    - Category Percolate
- Our Content API (Symfony) is responsible to call those indices, merge the results into defined response and send to our React Frontend
- Additionally, we use ES for monitoring (Logs of System, Webservers and Applications)

SETUP

- Installation is easy, either use vendor supplied repository for Debian/Redhat or Docker Container
- Even cookbooks for Ansible/Chef/Puppet are supplied by vendor
- https://www.elastic.co/guide/en/elasticsearch/reference/current/install-elasticsearch.html
- We have experienced various problems with updates with the same major version, e.g 6.4 to 6.5
  - Kibana Index problems ("Kibana gets stuck when upgrading from an older version") *
  - Scoring changed with 6.6 internally, we postponed upgrade to investigate the issue
  - Downgrading is most of the time not possible without breaking indices
  - We highly recommend to use methods like apt-pinning for make sure you don't get new version without testing

* https://www.elastic.co/guide/en/kibana/current/release-notes-6.5.0.html

INDEXING

- Schema free is possible, but ES recommends to use index mappings
- The index API adds or updates a typed JSON document in a specific index, making it searchable
  - Single or Bulk requests
- Calling refresh makes documents searchable (think commit in SQL)
- ES uses an inverted index https://speakerdeck.com/elasticsearch/stuff-a-search-engine-can-do?slide=25
- Use Alias Functionality for full export, create new index in background and switch alias on completion
  - "Renaming an alias is a simple remove then add operation within the same API. This operation is atomic, no need to worry about a short period of time where the alias does not point to an index"*

* https://www.elastic.co/guide/en/elasticsearch/reference/current/indices-aliases.html

INDEXING

- Nice side effect of alias is, that you can supply new mapping config and create new index with that new config
    - Live schema changes without any downtime (no ALTER TABLE!)
- Use filter wisely, decide if needed on query or index time or both
    - Stopwords
    - Stemmer
    - German Decompounder
    - German Normalization
    - Unique
    - HTML Strip etc..
- If you don't need to search over field, use type keyword (no filter)
- You can write your own filters
    - Mattress Size Token Filter

* https://www.elastic.co/guide/en/elasticsearch/reference/current/indices-aliases.html

QUERYING AND CACHING

- Generally, ES is fast, especially if you can run it as a cluster
  - Reason is, that the index is splitted into several parts (shards) and distributed over several servers (nodes)
- "By default, the requests cache will only cache the results of search requests where size=0, so it will not cache hits, but it will cache hits.total, aggregations, and suggestions.“
  - We splitted search call into 2 calls, one for searching with size > 0 and a second call for aggregations (statistics)
  - Second call can now be cached from ES
- https://www.ebayinc.com/stories/blogs/tech/elasticsearch-performance-tuning-practice-at-ebay/

* https://www.elastic.co/guide/en/elasticsearch/reference/current/shard-request-cache.html

TIPS AND TRICKS

- Never ever use ES as your primary data storage - Eventual Consistency
  - Eventual Consistency Synchronous doc based replication.
  - Get by ID may show delays up to 1 sec.
  - Eventual Consistency Synchronous doc based replication. Get by ID may show delays up to 1 sec. Configurable write consistency: one, quorum, allConfigurable write consistency: one, quorum, all
- Use Kibana for playing with queries
  - autocompletion helps there a lot
  - copy to curl :)

TIPS AND TRICKS

- Use Query Profiler in Kibana
- Export your query from your Jetbrains IDE with json_encode and use it in Kibana
- Raise your memory in ES, if you see messages like this
  - [2019-02-04T00:41:43,549][WARN][o.e.m.j.JvmGcMonitorService] [SF9DcSX] [gc] [1571159] overhead, spent [794ms] collecting in the last [1.4s]
- Lower shards to 1 if your index is < 1GB and you have only one server
- Raise refresh to > 1s to avoid small segments and performance penalty
- ES relies heavily on the filesystem cache of the OS. Always have enough RAM available, otherwise the index file will be read from Disc to RAM for each call..
  - "The filesystem cache will be used in order to buffer I/O operations. You should make sure to give at least half the memory of the machine running Elasticsearch to the filesystem cache."

* https://www.elastic.co/guide/en/elasticsearch/reference/current/tune-for-indexing-speed.html

FURTHER READINGS

- https://www.elastic.co/guide/en/elasticsearch/reference/current/index.html
- https://bonsai.io/blog/ideal-elasticsearch-cluster
- https://speakerdeck.com/elasticsearch/stuff-a-search-engine-can-do?slide=25
- https://www.ebayinc.com/stories/blogs/tech/elasticsearch-performance-tuning-practice-at-ebay/
- https://lucene.apache.org/core/index.html
- http://solr-vs-elasticsearch.com/
- http://www.elasticsearchtutorial.com/elasticsearch-in-5-minutes.html
- https://sematext.com/blog/solr-vs-elasticsearch-differences/

END

- Questions, Remarks, Concerns?