Elasticsearch

Frederick Cheung
CTO, dressipi.com

@fglc2 / spacevatican.org

Dressipi

- Your online virtual stylist
- Clothes that suit your shape
- Clothes that suit your preferences
- Many different filters to apply to garments

Our documents

- Garments (dresses, skirts, shoes, hats etc.)
- Objective attributes (price, brand, retailer etc)
- Retailer descriptions / names
- Curated feature sets: necklines, styles, fits, materials etc.

Our requirements

- Free text search (for users)
- Filtering (price, specific feature(s), brand, category...)
- Per user sort order based on personal preference
- Searchable per user collections: garment lists, liked garments etc

Previous solutions

- Mysql based: joins hell
- Hybrid mysql + solr - messy
- Per user collections in mongo with ordering from recommendation engine + filterable attributes

Elasticsearch

- Lucene based search engine (like solr): rich queries, facets etc.
- Spiritual child of Compass
- Great multi-index support
- Designed for distributed operation
- Evolving quickly

API

- Restful (ish)
- documents are JSON (dynamic schema)
- Queries are JSON documents - easy to nest/combine
- all you need is curl

```
{
  "query": {
    "custom_score": {
      "query": {
        "filtered": {
          "query": {"match_all": {}}
          "filter": {
            "and": [
              {"term": {"garment_category_id": 1}},
              {"range": {"price": {"from": 50, "to": 100}}}
            ]
          }
        },
        "script": "score_by_recommendedness",
        "params": {
          "profile_id": 12345
        },
        "lang": "native"
      }
    },
    "size": 15
}
```

Other options

- Sphinx (+ Thinking Sphinx)
- SOLR (+ sunspot)
- Amazon CloudSearch
- sql fulltext search/filtering

Versus sphinx

- better realtime search
- sphinx documents are flat (multi valued attributes exist, but only numerical)
- better distributed story
- sphinx search api not as rich as the lucene based solutions
- sphinx not as customizable (eg custom scoring)

Versus solr

- Both use lucene - fundamentally similar querying abilities
- Better HA / distributed story. Solrcloud looks fiddlier, more limited (and not yet done)
- better performance with heavy indexing load (http://engineering.socialcast.com/2011/05/realtime-search-solr-vs-elasticsearch/)
- I like the elasticsearch api better
- moving quicker

Versus cloudsearch

- Amazon's in the cloud search service
- whizzy autoscaling stuff - adds/resizes instances & reshards as needed
- more limited api
- wasn't available when we started out

Versus relational DB

- mysql fulltext search - gets slow quick
- fulltext engines built into dbs are less flexible
- Filtering & ordering on attributes spread across several tables gets nasty pretty quickly

Some elasticsearch highlights

Configuration

- Nearly everything exposed via API
- Index creation, schemas (mapping), index settings
- Very rarely need to fiddle with config files!
- Dynamic schema (not schema-less)

Deployment

- self contained - download and run
- bonsai.io provides hosted elasticsearch (also available as a heroku add-on)
- Easy sharding/replication: just an index creation parameter
- Node autodiscovery
- Cloud friendly via cloud-aws plugin

```
cloud:
  aws:
    access_key: AKXXXXXXXXXXXXXXXXSQ
    secret_key: Nv9xxxxxxxxxxxxxxxxxxxxxxxxx
    region: eu-west-1
discovery:
  type: ec2
  ec2:
    tag:
      role: elasticsearch
      deployment: "<%= env[:deployment_identifier] %>"
```

Parent/child

- Elasticsearch has the concept of a parent and child document
- Others need denormalization or dodgy workarounds
- has_child filter selects parents whose children match a query
- we use this to represent searchable per user lists

```
{
  "query": {
    "filtered": {
      "query": {
        "text": {"description": "red dress"}
      },
      "filter":{
        "has_child": {
          "type": "rating",
          "query" : {
            "filtered": {
             "query": { "match_all": {}},
             "filter" : {
               "and": [
                 {"term": {"user_id": 1234}},
                 {"range": {"rating": {"gt": 3}}}
               ]
             }
          }
        }
      }
```

Loads more details at

http://spacevatican.org/2012/6/3/fun-with-elasticsearch-s-children-and-nested-documents

Percolator

- ES allows you to register queries as percolators
- When you index a document it will optionally tell you which queries matched

Rivers

- Push or pull data sources for ES
- couchdb river hooks onto /_changes
- mongodb river reads replication oplog

Easy to extend

- custom_score, custom_filters_score: rank documents by script
- Not just single expressions! We solved our ordering problem with this
- http://spacevatican.org/2012/5/12/elasticsearch-native-scripts-for-dummies

- plugin api (thrift, rivers, cloud-aws, scripting languages, language specific analyzers etc.)
- Good intro to plugins http://jfarrell.github.com/

Ruby

- tire, rubberband, elastic_searchable, jruby-elasticsearch, https://github.com/Asquera/eson

- It's just JSON - hardly needed for simple stuff

```
filter :terms, :category_id => [1,2,3]

                vs

{filter: {terms: {category_id: [1,2,3]}}}
```

```
Garment.search do
  query { text :title, "red dress"}
  filter :term, :available => true
end
```

Returns fake garment objects - doesn't hit the db

```
Garment.search :load => true do
  query { text :title, "red dress"}
  filter :term, :available => true
end
```

Returns real garment objects

ES is a data store

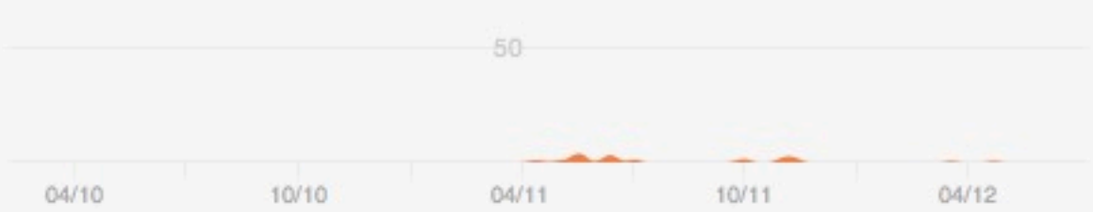
- eventually consistent document oriented data store
- Tire provides ActiveModel compliant support
- Example at https://github.com/fcheung/tire-blog

Some downsides

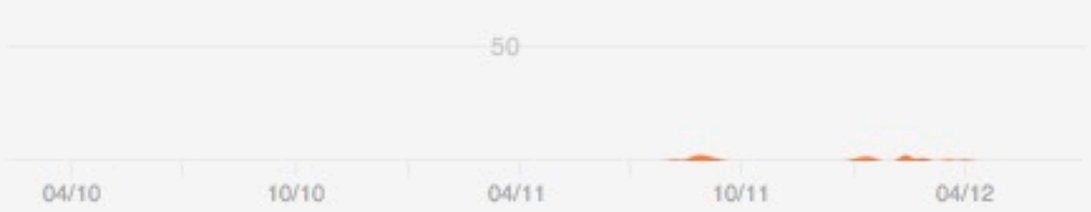
- Less commercial support etc (helpful mailing list - Shay Banon very active on it)
- Your hosting partner may not provide it
- low bus number

kimchy
3,384 commits / 1,230,868 ++ / 908,877 --

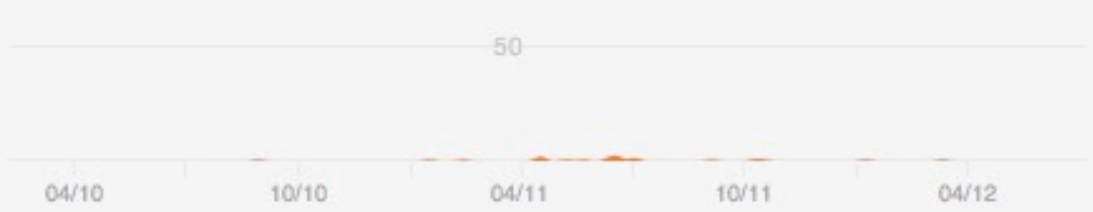
imotov
33 commits / 4,689 ++ / 467 --



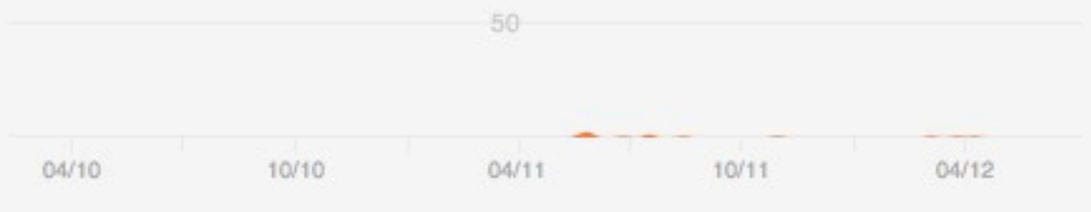
Paikan
22 commits / 2,869 ++ / 397 --



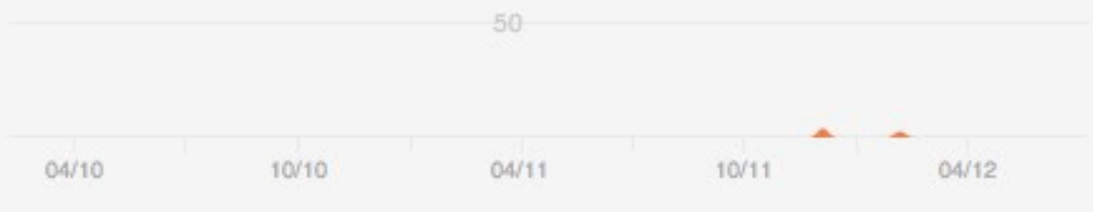
lukas-vlcek
20 commits / 8,707 ++ / 429 --



nlalevee
13 commits / 153 ++ / 62 --



karussell
10 commits / 992 ++ / 69 --



Shay is a legend

Questions?

- fred@dressipi.com
- @fglc2
- http://spacevatican.org