

From Java to Python:



Migrating Search Functionality at billiger.de



Patrick Schemitz, solute GmbH
pycon.de 2017

solute GmbH – billiger.de

- ca. 300 employees in Karlsruhe, Leipzig, Dresden and Plovdiv (Bulgaria)



==> price comparison <==

- **billiger.de** and syndication partner dataset:
 - 2500 data feeds (from 50000+ dealers)
 - 65+ mio offers
 - volatility ø 6 mio offers/day
 - ø 200000 visitors/day (billiger.de)
 - ø 17 mio search requests/day (total)



← → C Sicher | <https://www.billiger.de/search?searchstring=goodyear+winterreifen&filter=&search=1&stat=1>

billiger.de Kategorien Anmelden

Startseite > Suche nach "goodyear winterreifen"

Ihre Suche nach "goodyear winterreifen" liefert 249 Ergebnisse aus 32 Shops

Beliebte Filter [Alle Filter anzeigen](#)

Kategorie

- ☐ Winterreifen (191)
- ☐ LKW-Reifen (27)
- ☐ SUV-Reifen (29)

[Alle anzeigen](#)

Preis

34,00 € 520,00 € [>](#)

Marke

- ☐ Goodyear (249)

[Alle Filter anzeigen](#)

Passende Kategorien

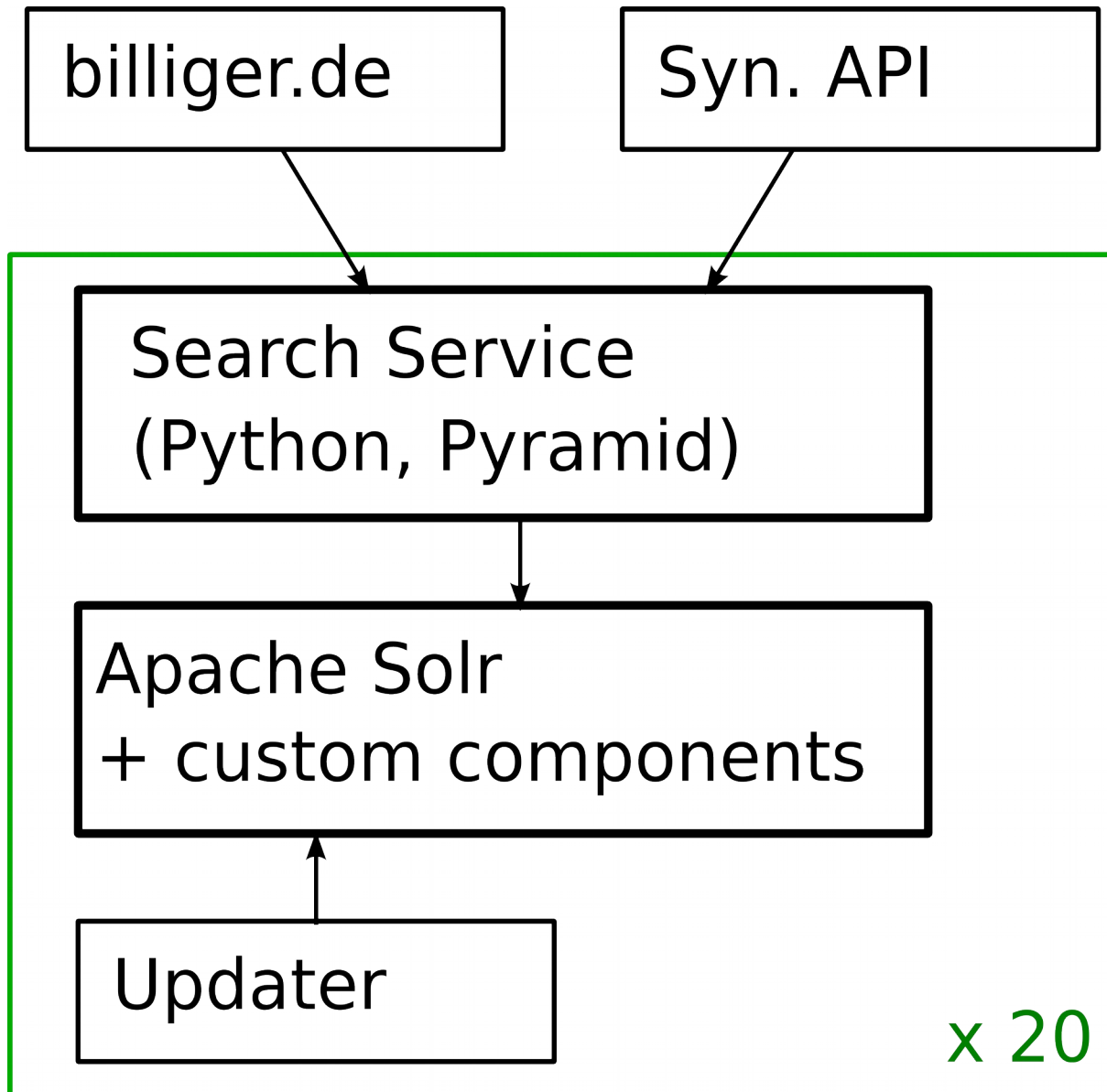
- Winterreifen von Goodyear
- PKW-Reifen von Goodyear

Sortieren nach: **Relevanz (relevantestes oben)** Ansicht:

<p>Goodyear UltraGrip 9 185/65 R15 88T in Winterreifen</p> <p>★★★★☆ (7) Gesamtnote 1,5 (gut)</p> <p>Reifenbreite 185 mm Querschnittsverhältnis 65 % Felgendurchmesser 15"</p> <p>67 dB</p>	<p>Produkt vergleichen</p> <p>ab 49,28 €* </p> <p>48 Preise vergleichen</p>
<p>Goodyear UltraGrip 8 165/65 R14 79T in Winterreifen</p> <p>★★★★☆ (4) Gesamtnote 2,0 (gut)</p> <p>Reifenbreite 165 mm Querschnittsverhältnis 65 % Felgendurchmesser 14"</p> <p>68 dB</p>	<p>Produkt vergleichen</p> <p>ab 50,67 €* </p> <p>45 Preise vergleichen</p>
<p>Goodyear Ultra Grip 9 155/65 R14 75T in Winterreifen</p> <p>★★★★☆ (4) Gesamtnote 2,0 (gut)</p>	<p>Produkt vergleichen</p>

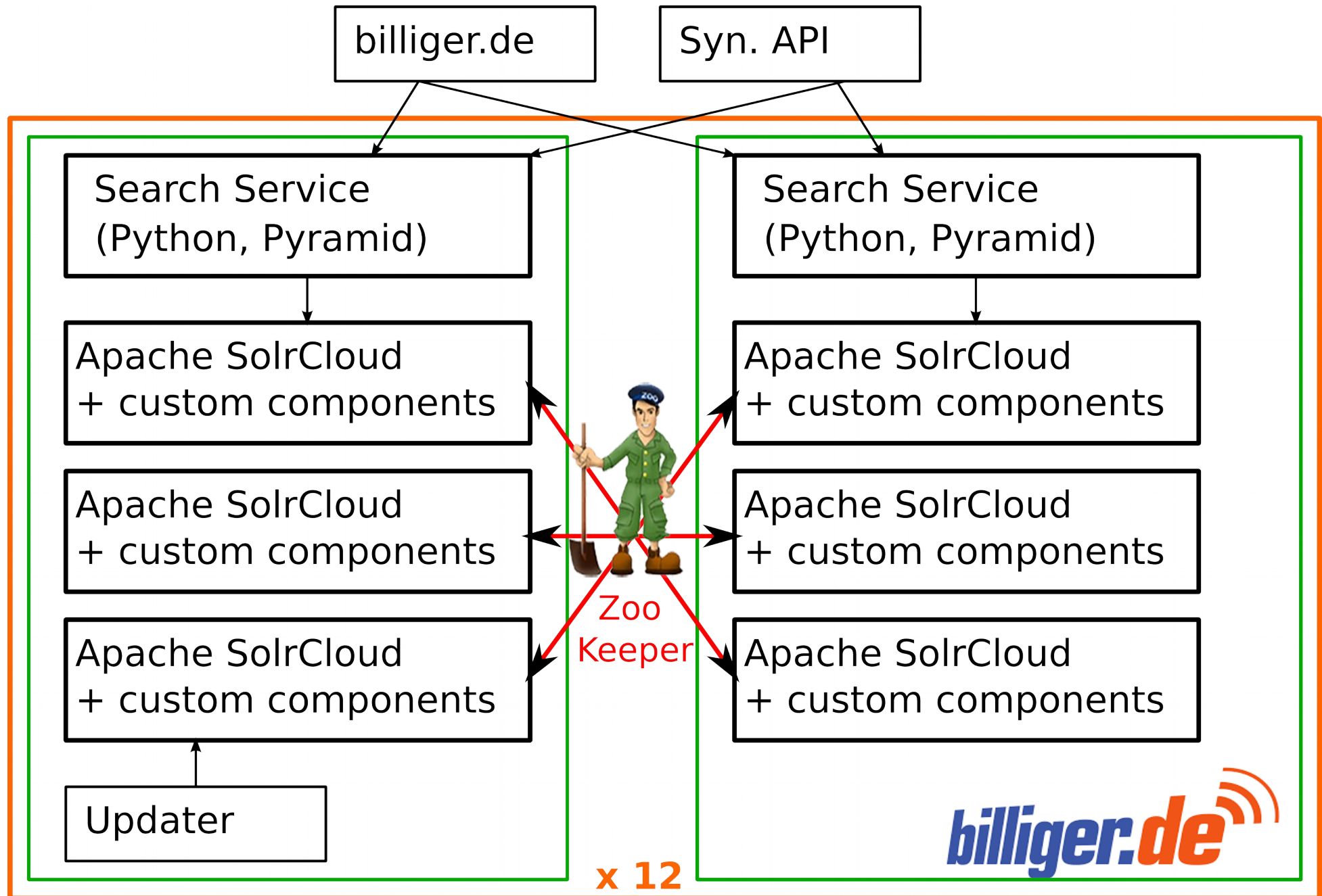
■ 17M queries/day ■ 65M offers ■

(Old) Architecture



- Dual 6-8 cores, 128 GB
- 1 Solr instance w/ *complete index*
- Search Service: *encapsulates actual search* (JSON/RPC, Python, Pyramid)

Target (Cluster) Architecture



Solr Components Interface

- Solr **SearchComponent** interface (*non-cluster*):

Before actual search (but after query parsing):

```
public abstract void prepare(  
    ResponseBuilder rb) throws IOException;
```

After actual search:

```
public abstract void process(  
    ResponseBuilder rb) throws IOException;
```

Custom Components I:

QLTB

- **Q**uery-**L**ocal **T**erm **B**oost: Boosting successful “terms” (offers, filters) for select queries:

```
<boosts>
  <query text="xperia x">
    <term boost="2.1" field="filter_brands"
      value="12316"/> <!--Sony-->
    <term boost="110.0" field="filter_categories"
      value="4373"/> <!--Handys ohne Vertrag-->
    <term boost="13000.0" field="id"
      value="1:808801927"/> <!--Xperia X 32GB black-->
  </query>
```

- Data harvested from clicklogs
- Updater puts QLTB.XML file into Solr config dir
- OSS: <https://github.com/solute/qltb>

QLTB Component

```
public class QLTBComponent extends SearchComponent {

    @Override
    public final void prepare(final ResponseBuilder rb) {
        Query query = rb.getQuery();
        String queryStr = rb.getQueryString();
        IndexReader reader = rb.req.getSearcher().getIndexReader();
        List<Query> boostTerms = getBoostsMap(reader,
                                                rb.req.getCore()).get(queryStr);
        BooleanQuery newq = new BooleanQuery(true);
        newq.add(query, BooleanClause.Occur.MUST);
        for (Query term : boostTerms) {
            newq.add(term, BooleanClause.Occur.SHOULD);
        }
        rb.setQuery(newq);
    }

    @Override
    public void process(final ResponseBuilder rb) {
    }
}
```


QLTB: Cluster Trouble

- Updater not on all nodes present
→ no qltb.xml?!
- SolrCloud: config distrib. by Apache ZooKeeper
- ZooKeeper config file size limit: 1 MB
(need to recompile to adapt)
- We have a (company) standard way to distribute files: MogileFS
- Move functionality to Search Service!



QLTB Code: Python “prepare”

```
def search(..., query, ...):  
    # ...  
    qltb_terms = qltb.get_boost_terms(query)  
    if qltb_terms:  
        query_boosts = filters_to_solr(qltb_terms, with_boosts=True)  
        for bq in query_boosts:  
            solr_request.append(("bq", bq))  
    # ...
```

QLTB Code: Python “prepare”

```
def search(..., query, ...):  
    # ...  
    qltb_terms = qltb.get_boost_terms(query)  
    if qltb_terms:  
        query_boosts = filters_to_solr(qltb_terms, with_boosts=True)  
        for bq in query_boosts:  
            solr_request.append(("bq", bq))  
    # ...
```

Fits great into existing code for filters and boosts:

```
def search(..., query, ..., filters, boosts, ...):  
    # ...  
    query_filters = filters_to_solr(filters, with_boosts=False)  
    for fq in query_filters:  
        solr_request.append(("fq", fq))  
  
    query_boosts = filters_to_solr(boosts, with_boosts=True)  
    for bq in query_boosts:  
        solr_request.append(("bq", bq))  
    # ...
```

Custom Components II:

Facetting & Filter Alternatives

Ihre Suche nach "apple iphone" liefert 199 Ergebnisse aus 50 Shops

Sortieren nach: **Relevanz (relevantestes oben)** Ansicht:   

Handys ohne Vertrag 

[Alle Filter zurücksetzen](#) 

Beliebte Filter

 Alle Filter anzeigen

Reduzierte Artikel 

☐ % **SALE** (1)

Kategorie 1 

☒ Handys ohne Vertrag (199)

☐ Handytaschen (52.673)

[Alle anzeigen](#)

[Filter zurücksetzen](#)

Preis 

	<p>Apple iPhone SE 32GB spacegray in Handys ohne Vertrag</p> <p>★★★★☆ (12) Gesamtnote 1,7 (gut)</p> <p>Display-Diagonale 4"</p> <p>Integrierter Speicher 32 GB</p> <p>Akkukapazität 1620 mAh</p>	<p> Produkt vergleichen</p> <p>ab 250,50 €*</p> <p>39 Preise vergleichen</p>
	<p>Apple iPhone 7 32GB schwarz in Handys ohne Vertrag</p> <p>★★★★★ (31) Gesamtnote 1,2 (sehr gut)</p>	<p> Produkt vergleichen</p>

Facetting & Filter Alternatives

Ihre Suche nach "apple iphone" liefert 166 Ergebnisse aus 50 Shops

Sortieren nach: **Relevanz (relevantestes oben)** Ansicht:   

Handys ohne Vertrag X 285,00 € - 1.319,00 € X [Alle Filter zurücksetzen X](#)

Beliebte Filter

Alle Filter anzeigen

Reduzierte Artikel

☐ % SALE (1)

Kategorie 1

☒ Handys ohne Vertrag (166)

☐ Handys mit Vertrag (2)

☐ Kopfhörer (3)

Alle anzeigen

Filter zurücksetzen

Preis 1

285,00 € 1.319,00 € >

	<p>Apple iPhone 7 32GB schwarz in Handys ohne Vertrag</p> <p>★★★★★ (31) Gesamtnote 1,2 (sehr gut)</p> <p>Display-Diagonale 4.7" Integrierter Speicher 32 GB Akkukapazität 1960 mAh</p>	<p>Produkt vergleichen</p> <p>ab 575,00 €*</p> <p>52 Preise vergleichen</p>
	<p>Apple iPhone 6 32GB spacegrau in Handys ohne Vertrag</p> <p>★★★★☆ (11) Gesamtnote 2,0 (gut)</p> <p>Display-Diagonale 4.7" Integrierter Speicher 32 GB Akkukapazität 1810 mAh</p>	<p>Produkt vergleichen</p> <p>ab 369,00 €*</p> <p>31 Preise vergleichen</p>

Filter Alternatives Component

```
@Override public final void process(final ResponseBuilder rb) {
    //...
```

```
    for (String filterField : termFilterFields) {
```

```
        List<Query> remainingFilters = new ArrayList<Query>();
        List<String> filterFields = new ArrayList<String>();
```

```
        recreateQueryFields(
```

```
            List<Query> alternativ
            alternativ
            DocSet dcs
```

```
            UnInverted
            filter
```

```
            NamedList<
            limit,
```

```
            NamedList<
            for (Entry
```

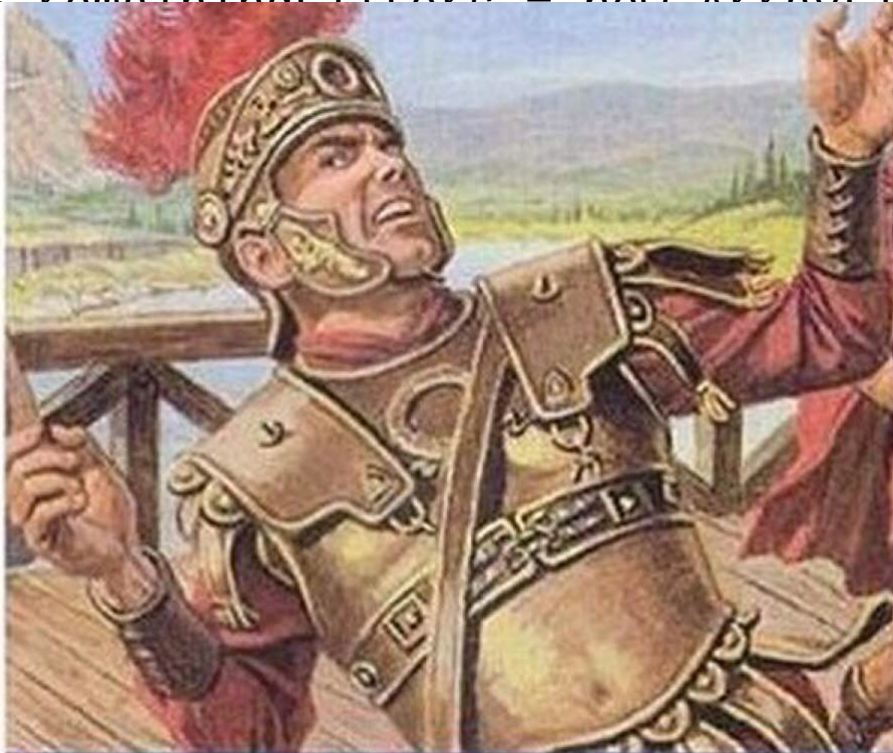
```
                String
                String
```

```
                if ((
```

```
                    .equals(origFilterField)) {
```

```
                        fieldCounts.add(keyValue[1], valuePair.getValue());
```

```
                    }
```



**ABSOLUTELY
BARBARIC**

```
        filterFields,
        rs);
        List<Query>()
```

```
;
        filters);
```

```
        InvertedField(
        searcher, dcs, 0,
```

```
        List<Integer>();
        ts) {
            ey();
            it(":");
```


Filter Alternatives Component

```

@Override public final void process(final ResponseBuilder rb) {
    //...
    for (String filterField : termFilterFields) {
        List<Query> remainingFilters = new ArrayList<Query>();
        List<String> filterFields = new ArrayList<String>();
        filterFields.add(filterField);
        recreateQueriesWithoutFields(filters, filterFields,
                                    remainingFilters);

        List<Query> alternativeFilters = new ArrayList<Query>();
        alternativeFilters.add(query);
        alternativeFilters.addAll(remainingFilters);
        DocSet dcs = searcher.getDocSet(alternativeFilters);
        UnInvertedField uif = UnInvertedField.getUnInvertedField(
            filterField, rb.req.getSearcher());
        NamedList<Integer> counts = uif.getCounts(searcher, dcs, 0,
            limit, 1, false, "count", null);
        NamedList<Integer> fieldCounts = new NamedList<Integer>();
        for (Entry<String, Integer> valuePair : counts) {
            String filterKeyValue = valuePair.getKey();
            String[] keyValue = filterKeyValue.split(":");
            if (("filter_" + keyValue[0])
                .equals(origFilterField)) {
                fieldCounts.add(keyValue[1], valuePair.getValue());
            }
        }
    }
}

```


Filter Alternatives: Cluster Trouble

- **UnInvertedField** dropped in Solr 6
- `getDocSet()` (=search) in a loop → **latency!**
- Move functionality to Search Service!

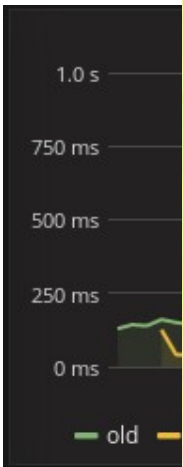
Filter Alternatives in Python

```
def search(index, query, page_size, page_no,
           sort_mode, filters, boosts, options, ...):
    # ...
    alternatives_requests = []
    for filter_key, filter_values in filters.iteritems():
        partial_filters = copy.deepcopy(filters)
        partial_filters.pop(filter_key)
        alternatives_requests.append((
            index, query,
            0, 0, # paging irrelevant here, 0 is fastest
            sort_mode,
            partial_filters,
            {}, # boosts irrelevant here, none are fastest
            options,
        ))

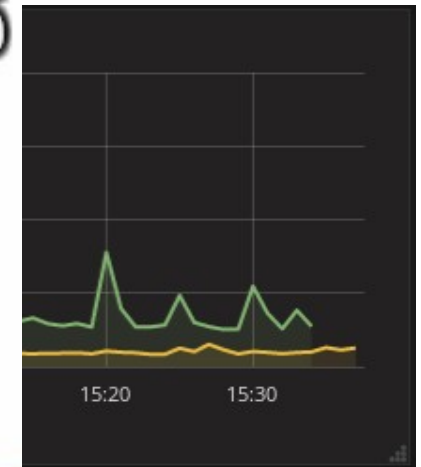
    alternatives_responses = search_threaded(
        alternatives_requests,
        _pool
    )
```

Conclusion

- Switch



**JAVA SEARCH
COMPONENTS**



- SolrC
- Encap
- to Sol
- Moved
- Python

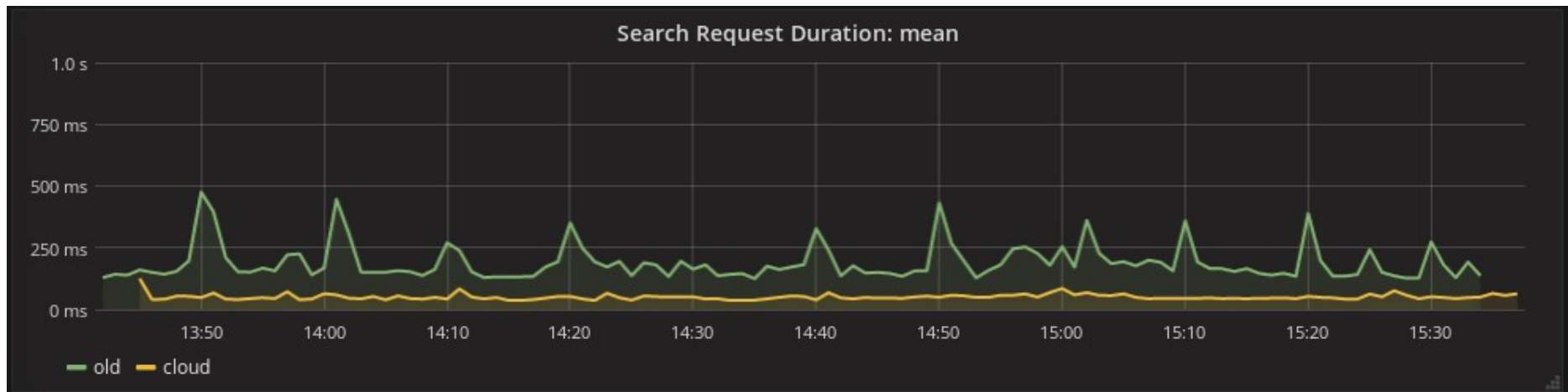


**PYTHON
SEARCH
SERVICE**

.I in migrating

Conclusion

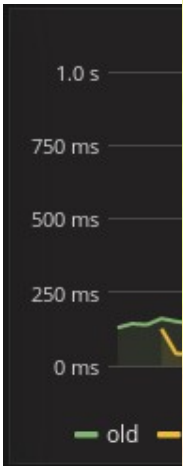
- Switch from Solr to SolrCloud ✓



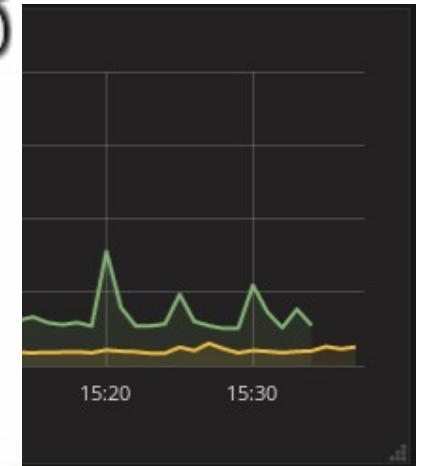
- SolrCloud works great ✓
- Encapsulating Solr in a (Python) service **crucial** in migrating to SolrCloud ✓
- Moved six SearchComponents to Python ✓
- Python is **awesome** ✓

Conclusion

- Switch



**JAVA SEARCH
COMPONENTS**



- SolrC
- Encap
- to Sol
- Move
- Pytho



**PYTHON
SEARCH
SERVICE**

.I in migrating

Minimum Match Python Code

```
def search(..., query, ...):  
    # ...  
    solr_request.append(("mm", "100%"))  
    # ...  
    result = solr_search(url, solr_request, shop_fq, options)  
    # ...  
    if result["total_hits"] == 0:  
        _replace_param(solr_request, "mm", "75%")  
        result = solr_search(url, solr_request, shop_fq, options)
```

Minimum Match: Cluster Trouble

- prepare() executed on each shard/instance:
 - getDocSet() (= *search*) on each shard:
 - **mm might vary between shards**

Some shards would return precise hits, others imprecise
- Move functionality to Search Service!

Minimum Match Component

```
public class FuzzyComponent extends SearchComponent {  
  
    @Override public void prepare(ResponseBuilder rb) {  
        String queryString = rb.getQueryString();  
        SolrParams params = rb.req.getParams();  
        Query q = rb.getQuery();  
        List<Query> filters = rb.getFilters();  
        List<Query> queries = new ArrayList<Query>();  
        queries.add(q);  
        queries.addAll(filters);  
        DocSet result = rb.req.getSearcher().getDocSet(queries);  
        if (result.size() > 0)  
            return;  
        q = QParser.getParser("{!mm=75%} " + queryString,  
                               defType, rb.req).getQuery();  
        queries = new ArrayList<Query>();  
        queries.add(q);  
        queries.addAll(filters);  
        DocSet result = rb.req.getSearcher().getDocSet(queries);  
        if (result.size() > 0)  
            rb.setQuery(q);  
    }  
}
```

Custom Components III:

“Minimum Match” (fuzzy search)

- “Minimum Match” parameter: number (or percentage) of words in a query that need to match a doc.
- Example: query “apple samsung galaxy s7”
 - mm=100% (*all* terms must match) yields no hits
 - mm=75% (3 terms must match) yields “samsung galaxy s7” but also **random stuff** (if any): “apple samsung s7”, “apple galaxy s7” ...
 - Strategy:
 - try mm=100% first
 - try mm=75% only if no mm=100% hits