

Search Evolution – von Lucene zu Solr und ElasticSearch



Florian Hopf

@fhopf

<http://www.florian-hopf.de>



04.07.2013

Elasticsearch

| | |
|---------------------|--|
| Datum | : 12.06.13 |
| Ort | : <u>Uni KA, HS101</u> |
| Speaker | : Alexander Reelsen, Elasticsearch |
| Homepage | : <u>http://elasticsearch.org/</u> |
| Promo-Flyer | : - |
| Slides | : <u>Slideshare</u> |
| Bilder | : - |
| Aufzeichnung | : - |
| Zus. Links | : - |

Eine performante, verteilte und relevante Suchmaschine zu erstellen und zu verwalten, ist eine komplexe Aufgabe. Dieser Vortrag zeigt die auftretenden Komplexitäten beim Aufsetzen einer verteilten Suchmaschine - aus Administrator-, Entwickler- und Usersicht - aber auch, wie man einige dieser Probleme mit Elasticsearch als Plattform lösen kann. Des Weiteren werden ebenfalls die initiale Konfiguration, Integrationsszenarien, Betrieb in der Produktionsumgebung sowie die mögliche Erweiterbarkeit von Elasticsearch beleuchtet.

Alexander Reelsen ist Software Engineer bei Elasticsearch. Nach mehreren Jahren im Bereich der JVM basierten Webentwicklung mit Fokus auf leicht zu wartenden und zu entwickelnden Frameworks wie dem Play Framework oder Dropwizard liegt sein Fokus aktuell im Bereich Information Retrieval.

Lucky Seven

Nach der Veröffentlichung vom JDK 7 und dem darauf folgenden JDK 8 in 2012 sollen die angekündigten Inhalte für **Java** SE 7 und **Java** SE 8 genauer betrachtet werden. Im JDK 7 geht es um kleinere Sprachverbesserungen im Project Coin, die Unterstützung für dynamisch typisierte Sprachen ("InvokeDynamic") JSR 292, Concurrency und Collections Updates inklusive Fork/Join Framework. Der Vortrag wird sich mit dem aktuellen Stand von **Java** 7 beschäftigen, die wichtigsten Änderungen, Erweiterungen sowie

Kategorien: JDK

Apache Wicket

Apache Wicket ist ein komponentenbasiertes Web-Framework, welches als Top-Level Projekt unter dem Dach der Apache Foundation entwickelt wird. Im Gegensatz zu anderen Web-Frameworks verzichtet Wicket konsequent auf Konfigurationsdateien und setzt auf Objekt-Orientierung, reines HTML und **Java**..., die wiederverwendbar sind. Gleiches gilt für den Einsatz von **Java** Script Bibliotheken und AJAX. Im Rahmen des Vortrags wird erläutert wie **Java** basierte Web-Anwendungen mit Hilfe von Apache Wicket

Kategorien: Web

Integration ganz einfach mit Apache Camel

Apache Camel ist ein beliebtes Integrationsframework der Apache Foundation. Grundlage sind die von Gregor Hohpe im gleichnamigen Buch beschriebenen Enterprise Integration Patterns. Mit Hilfe von Domain Specific Languages in **Java** oder XML können aus Integration Patterns Integrationsabläufe modelliert und direkt ausgeführt werden. Eine große Anzahl Komponenten ermöglicht, verschiedenste Formate... können als Standalone **Java** Applikation, WAR Archiv oder OSGi bundle deployed werden. Der Vortrag gibt

Kategorien: Architektur Integration

Programmiersprachen - Zentrale Architekturentscheidung oder unwichtiges Detail?

Lange Zeit war die leichteste Entscheidung in einem Projekt die über die einzusetzende Programmiersprache - denn sie wurde in aller Regel schon längst auf Unternehmensebene getroffen. Für viele Entwickler und Architekten in Großunternehmen war **Java** dabei für mehr als ein Jahrzehnt die offensichtliche Wahl. In letzter Zeit gewinnen diverse Alternativen (wie Scala, JRuby, Groovy oder Clojure) mehr und mehr an Popularität. Dass eine davon **Java** den Rang ablaufen wird, ist ebenso unwahrscheinlich

Kategorien: Architektur JRuby Clojure Groovy JVM

Index

Elasticsearch

Datum : 12.06.13
Ort : [Uni KA_HS101](#)
Speaker : Alexander Reelsen, Elasticsearch
Homepage : <http://elasticsearch.org/>
Promo-Flyer : -
Slides : [Slideshare](#)
Bilder : -
Aufzeichnung : -
Zus. Links : -

Eine performante, verteilte und relevante Suchmaschine zu erstellen und zu verwalten, ist eine komplexe Aufgabe. Dieser Vortrag zeigt die auftretenden Komplexitäten beim Aufsetzen einer verteilten Suchmaschine - aus Administrator-, Entwickler- und Usersicht - aber auch, wie man einige dieser Probleme mit Elasticsearch als Plattform lösen kann. Des Weiteren werden ebenfalls die initiale Konfiguration, Integrationszenarien, Betrieb in der Produktionsumgebung sowie die mögliche Erweiterbarkeit von Elasticsearch beleuchtet.

Alexander Reelsen ist Software Engineer bei Elasticsearch. Nach mehreren Jahren im Bereich der JVM basierten Webentwicklung mit Fokus auf leicht zu wartenden und zu entwickelnden Frameworks wie dem Play Framework oder Dropwizard liegt sein Fokus aktuell im Bereich Information Retrieval.

Indizieren

Index

Suchen

Lucky Seven
Nach der Veröffentlichung von JDK 7 und dem darauf folgenden JDK 8 in 2012 sollen die angekündigten Inhalte für Java SE 7 und Java SE 8 genauer betrachtet werden. Im JDK 7 geht es um kleinere Sprachverbesserungen im Project Coin, die Unterstützung für dynamisch typisierte Sprachen ("invokeDynamic"), JSR 290, Concurrency und Collections Updates inklusive Fork/Join Framework. Der Vortrag wird sich mit dem aktuellen Stand von Java 7 beschäftigen, die wichtigsten Änderungen, Erweiterungen sowie
Kategorien: JDK

Apache Wicket
Apache Wicket ist ein komponentenbasiertes Web-Framework, welches als Top-Level Projekt unter dem Dach der Apache Foundation entwickelt wird. Im Gegensatz zu anderen Web-Frameworks verzichtet Wicket konsequent auf Konfigurationsdateien und setzt auf Objekt-Orientierung, reines HTML und Java ... die wiederverwendbar sind. Gleiches gilt für den Einsatz von Java Script Bibliotheken und AJAX. Im Rahmen des Vortrags wird erläutert wie Java basierte Web-Anwendungen mit Hilfe von Apache Wicket
Kategorien: Web

Integration ganz einfach mit Apache Camel
Apache Camel ist ein beliebtes Integrationsframework der Apache Foundation. Grundlage sind die von Gregor Hohpe im gleichnamigen Buch beschriebenen Enterprise Integration Patterns. Mit Hilfe von Domain Specific Languages in Java oder XML können aus Integration Patterns Integrationsmuster modelliert und direkt ausgedrückt werden. Eine große Anzahl Komponenten ermöglicht, verschiedenste Formate ... können als Standalone Java Application, WAR Archiv oder OSGi bundle deployed werden. Der Vortrag gibt
Kategorien: Architektur Integration

Programmiersprachen - Zentrale Architekturentscheidung oder unwichtiges Detail?
Lange Zeit war die leichteste Entscheidung in einem Projekt die über die einzusetzende Programmiersprache - denn sie wurde in aller Regel schon längst auf Unternehmensebene getroffen. Für viele Entwickler und Architekten in Großunternehmen war Java dabei für mehr als ein Jahrzehnt die offensichtlichste Wahl, in letzter Zeit gewinnen diverse Alternativen wie Scala, JRuby, Groovy oder Clojure mehr und mehr an Popularität. Dass eine davon Java den Rang ablaufen wird, ist ebenso unwahrscheinlich
Kategorien: Architektur JRuby Clojure Groovy JVM

Index

Analyzing

Such
Evolution -
Von Lucene
zu Solr und
ElasticSearch

Verteiltes
Suchen mit
Elasticsearch

Analyzing

Such
Evolution -
Von Lucene
zu Solr und
ElasticSearch

1. Tokenization →

Verteiltes
Suchen mit
Elasticsearch

| Term | Document Id |
|---------------|-------------|
| Such | 1 |
| Evolution | 1 |
| Von | 1 |
| Lucene | 1 |
| zu | 1 |
| Solr | 1 |
| und | 1 |
| ElasticSearch | 1 |
| Verteiltes | 2 |
| Suchen | 2 |
| mit | 2 |
| Elasticsearch | 2 |

Analyzing

Such
Evolution -
Von Lucene
zu Solr und
ElasticSearch

1. Tokenization →

2. Lowercasing →

| Term | Document Id |
|---------------|-------------|
| such | 1 |
| evolution | 1 |
| von | 1 |
| lucene | 1 |
| zu | 1 |
| solr | 1 |
| und | 1 |
| elasticsearch | 1,2 |
| verteiltes | 2 |
| suchen | 2 |
| mit | 2 |

Verteiltes
Suchen mit
Elasticsearch

Analyzing

Such
Evolution -
Von Lucene
zu Solr und
ElasticSearch

1. Tokenization

2. Lowercasing

3. Stemming

| Term | Document Id |
|---------------|-------------|
| such | 1,2 |
| evolution | 1 |
| von | 1 |
| luc | 1 |
| zu | 1 |
| solr | 1 |
| und | 1 |
| elasticsearch | 1,2 |
| verteilt | 2 |
| mit | 2 |

Verteiltes
Suchen mit
Elasticsearch

Lucene

Inverted Index



Analyzer



Query Syntax

datenbank OR DB

title:elasticsearch

"apache lucene"

speaker:hopp~

elastic* AND date:[20130101 TO 20130501]

Relevance



Documents

Document

title

Such-Evolution

date

20130620

speaker

Florian Hopf

Attributes

Store

YES

NO

Attributes

Index

ANALYZED

YES

NO

Attributes

Index

ANALYZED

YES

NO

`TextField`

`StringField`

`StoredField`

Indexing

```
Document doc = new Document();
doc.add(new TextField(
    "title",
    "Suchen und Finden mit Lucene und Solr",
    Field.Store.YES));
doc.add(new StoredField(
    "speaker",
    "Florian Hopf"));
doc.add(new StringField(
    "date",
    "20120704",
    Field.Store.YES));
```

Indexing

```
Directory dir = FSDirectory.open(  
    new File("/tmp/testindex"));  
IndexWriterConfig config = new IndexWriterConfig(  
    Version.LUCENE_43,  
    new GermanAnalyzer(Version.LUCENE_43));  
IndexWriter writer = new IndexWriter(dir, config);  
  
writer.addDocument(doc);  
  
writer.commit();
```

Searching

```
QueryParser parser = new QueryParser(  
    Version.LUCENE_43,  
    "title",  
    new GermanAnalyzer(Version.LUCENE_43));  
Query query = parser.parse("suche");
```

Searching

```
IndexReader reader = IndexReader.open(dir);
IndexSearcher searcher = new IndexSearcher(reader);

TopDocs result = searcher.search(query, 10);
assertEquals(1, result.totalHits);

int id = result.scoreDocs[0].doc;
Document doc = searcher.doc(id);
String title = doc.get("title");
assertEquals(
    "Suchen und Finden mit Lucene und Solr",
    title);
```

Lucky Seven

Nach der Veröffentlichung vom JDK 7 und dem darauf folgenden JDK 8 in 2012 sollen die angekündigten Inhalte für **Java** SE 7 und **Java** SE 8 genauer betrachtet werden. Im JDK 7 geht es um kleinere Sprachverbesserungen im Project Coin, die Unterstützung für dynamisch typisierte Sprachen ("InvokeDynamic") JSR 292, Concurrency und Collections Updates inklusive Fork/Join Framework. Der Vortrag wird sich mit dem aktuellen Stand von **Java** 7 beschäftigen, die wichtigsten Änderungen, Erweiterungen sowie

Kategorien: JDK

Apache Wicket

Apache Wicket ist ein komponentenbasiertes Web-Framework, welches als Top-Level Projekt unter dem Dach der Apache Foundation entwickelt wird. Im Gegensatz zu anderen Web-Frameworks verzichtet Wicket konsequent auf Konfigurationsdateien und setzt auf Objekt-Orientierung, reines HTML und **Java**..., die wiederverwendbar sind. Gleiches gilt für den Einsatz von **Java** Script Bibliotheken und AJAX. Im Rahmen des Vortrags wird erläutert wie **Java** basierte Web-Anwendungen mit Hilfe von Apache Wicket

Kategorien: Web

Integration ganz einfach mit Apache Camel

Apache Camel ist ein beliebtes Integrationsframework der Apache Foundation. Grundlage sind die von Gregor Hohpe im gleichnamigen Buch beschriebenen Enterprise Integration Patterns. Mit Hilfe von Domain Specific Languages in **Java** oder XML können aus Integration Patterns Integrationsabläufe modelliert und direkt ausgeführt werden. Eine große Anzahl Komponenten ermöglicht, verschiedenste Formate... können als Standalone **Java** Applikation, WAR Archiv oder OSGi bundle deployed werden. Der Vortrag gibt

Kategorien: Architektur Integration

Programmiersprachen - Zentrale Architekturentscheidung oder unwichtiges Detail?

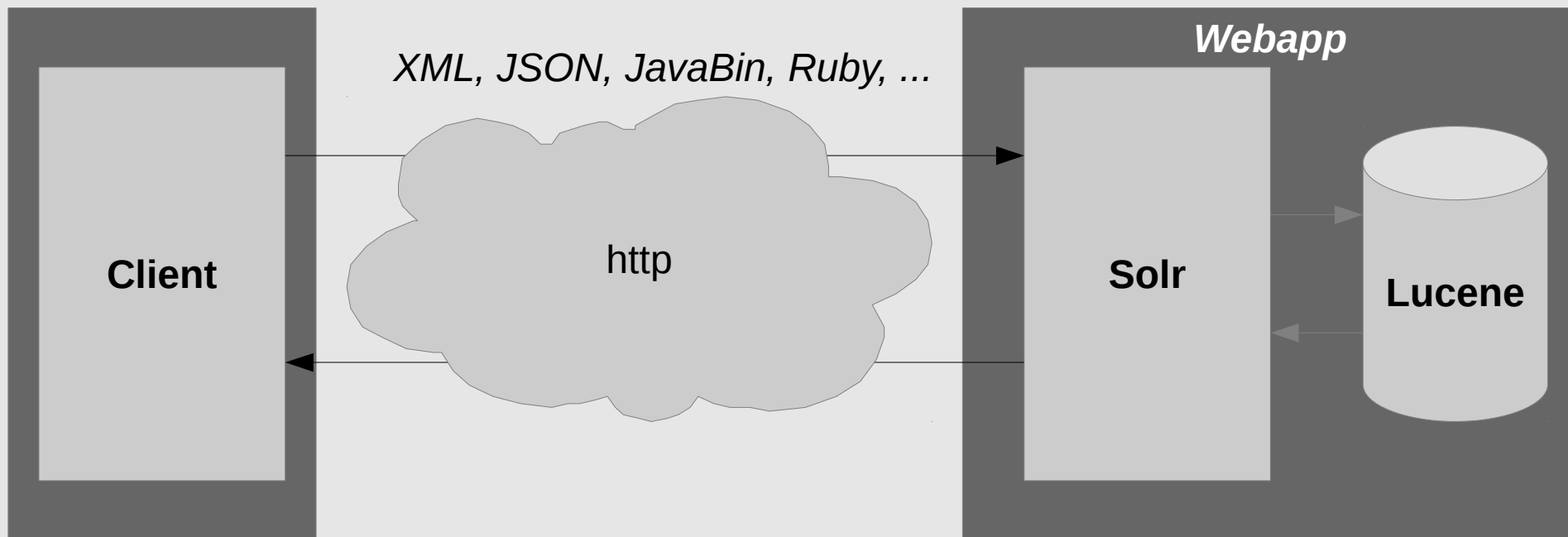
Lange Zeit war die leichteste Entscheidung in einem Projekt die über die einzusetzende Programmiersprache - denn sie wurde in aller Regel schon längst auf Unternehmensebene getroffen. Für viele Entwickler und Architekten in Großunternehmen war **Java** dabei für mehr als ein Jahrzehnt die offensichtliche Wahl. In letzter Zeit gewinnen diverse Alternativen (wie Scala, JRuby, Groovy oder Clojure) mehr und mehr an Popularität. Dass eine davon **Java** den Rang ablaufen wird, ist ebenso unwahrscheinlich

Kategorien: Architektur JRuby Clojure Groovy JVM

Apache

Solr





Schema

schema.xml

Field Types

Fields

Schema

```
<fieldType name="text_de" class="solr.TextField">
  <analyzer>
    <tokenizer
      class="solr.StandardTokenizerFactory"/>
    <filter
      class="solr.LowerCaseFilterFactory"/>
    <filter
      class="solr.GermanLightStemFilterFactory"/>
  </analyzer>
</fieldType>
```

Schema

```
<fields>
  <field name="title" type="text_de"
    indexed="true" stored="true"/>
  <field name="speaker" type="string"
    indexed="true" stored="true"
    multiValued="true"/>
  <field name="date" type="date"
    indexed="true" stored="true"/>
  [...]
</fields>
```

Indexing

```
SolrServer server =  
    new HttpSolrServer("http://localhost:8080");  
  
SolrInputDocument document =  
    new SolrInputDocument();  
document.addField("title",  
    "Suchen und Finden mit Lucene und Solr");  
document.addField("speaker", "Florian Hopf");  
  
server.add(document);  
server.commit();
```

Searching

```
SolrQuery solrQuery = new SolrQuery("suche");
solrQuery.setRequestHandler("/jug");

QueryResponse response = server.query(solrQuery);
assertEquals(1, response.getResults().size());

SolrDocument result = response.getResults().get(0);
assertEquals(
    "Suchen und Finden mit Lucene und Solr",
    result.get("title"));
assertEquals(
    "Florian Hopf",
    result.getFirstValue("speaker"));
```

Nach Datum sortieren

Suchen

Kategorien:

- [Web](#) (4)
- [Architektur](#) (3)
- [Agile](#) (1)
- [BPM](#) (1)
- [Clojure](#) (1)
- [Groovy](#) (1)
- [Integration](#) (1)
- [JCR](#) (1)
- [JDK](#) (1)
- [JRuby](#) (1)
- [JVM](#) (1)
- [Mobile](#) (1)
- [OSGi](#) (1)
- [REST](#) (1)
- [Spring](#) (1)

Speaker:

- [Florian Hopf](#) (2)
- [Bernd Rücker](#) (1)
- [Christian Schneider](#) (1)
- [Claus Augusti](#) (1)
- [Daniel Kurka](#) (1)
- [Michael Plöb](#) (1)
- [Oliver Gierke](#) (1)
- [Stefan Tilkov](#) (1)
- [Wolfgang Weigend](#) (1)

Datum:

- [2011](#) (5)
- [2012](#) (4)
- [2013](#) (1)

Lucky Seven

Inhalte für *Java* SE 7 und *Java* SE 8 genauer betrachtet werden. Im JDK 7 geht es um kleinere ...

Kategorien: [JDK](#)

Apache Wicket

konsequent auf Konfigurationsdateien und setzt auf Objekt-Orientierung, reines HTML und *Java*. Des Weiteren ...

Kategorien: [Web](#)

Integration ganz einfach mit Apache Camel

Specific Languages in *Java* oder XML können aus Integration Patterns Integrationsabläufe modelliert und ...

Kategorien: [Architektur](#) [Integration](#)

Programmiersprachen - Zentrale Architekturentscheidung oder unwichtiges Detail?

und Architekten in Großunternehmen war *Java* dabei für mehr als ein Jahrzehnt die offensichtliche Wahl ...

Kategorien: [Architektur](#) [JRuby](#) [Clojure](#) [Groovy](#) [JVM](#)

Apache Sling and JCR

Standardaufgaben radikal. Das im *Java* Community Process entwickelte JCR (Content Repository for *Java* Technology API ...

Kategorien: [Web](#) [REST](#) [OSGi](#) [JCR](#)

Huch, wo ist meine Architektur hin

Grundlage für langlebige *Java*-Applikationen zu legen, sowie eine Möglichkeit mit Spring lose gekoppelte ...

Kategorien: [Architektur](#) [Spring](#)

Suchen und Finden mit Lucene und Solr

-Facto-Standard im *Java*-Umfeld dar. Neben der Bereitstellung eines invertierten Index zur Datenablage ...

Kategorien: [Web](#)

Faceting

...

```
solrQuery.setFacet(true);  
solrQuery.addFacetField("speaker");  
  
QueryResponse response = server.query(solrQuery);  
List<FacetField.Count> speakerFacet =  
    response.getFacetField("speaker").getValues();  
assertEquals(1, speakerFacet.get(0).getCount());  
assertEquals("Florian Hopf",  
    speakerFacet.get(0).getName());
```




elasticsearch.

Indexing

```
curl -XPOST \
  'http://localhost:9200/jug/talk/' -d '{
    "speaker" :
      "Florian Hopf",
    "date" :
      "2012-07-04T19:15:00",
    "title" :
      "Suchen und Finden mit Lucene und Solr"
  }'

{"ok":true,"_index":"jug","_type":"talk",
"_id":"CeltdivQRGSvLY_dBZv1jw","_version":1}
```

Mapping

```
curl -XPUT \
  'http://host/jug/talk/_mapping' -d '{
  "talk" : {
    "properties" : {
      "title" : {
        "type" : "string",
        "analyzer" : "german"
      }
    }
  }
}'
```

Searching

```
curl -XGET \
  'http://host/jug/talk/_search?q=title:suche'
{...},
"hits":{"total":1,"max_score":0.054244425,
  "hits":[{"
    ...,
    "_score":0.054244425,
    "_source" : {
      "speaker" :
        "Florian Hopf",
      "date" :
        "2012-07-04T19:15:00",
      "title":
        "Suchen und Finden mit Lucene und Solr"
    }
  ]
}
```

Searching

```
curl -XGET  
'http://localhost:9200/jug/talk/_search' -d '{  
  "query" : {  
    "query_string" : {"query" : "suche"}  
  },  
  "facets" : {  
    "tags" : {  
      "terms" : {"field" : "speaker"}  
    }  
  }  
}'
```

Verteilung

ElasticSearch Connect **Phat** **cluster health: green (2, 5)**

Overview Browser Structured Query [+] Any Request [+]

Cluster Overview New Index

Moonstar, Danielle 2EhZOmz9S66xliHEX9bp3w
inet[/172.28.100.89:9200]
Info Actions

Phat 9l8LnzHkQBmpGJ_XPo7g6Q
inet[/172.28.100.89:9202]
Info Actions

jug
size: 89.4kb (178.8kb)
docs: 22 (22)
Info Actions

0

1

2

3

4

0

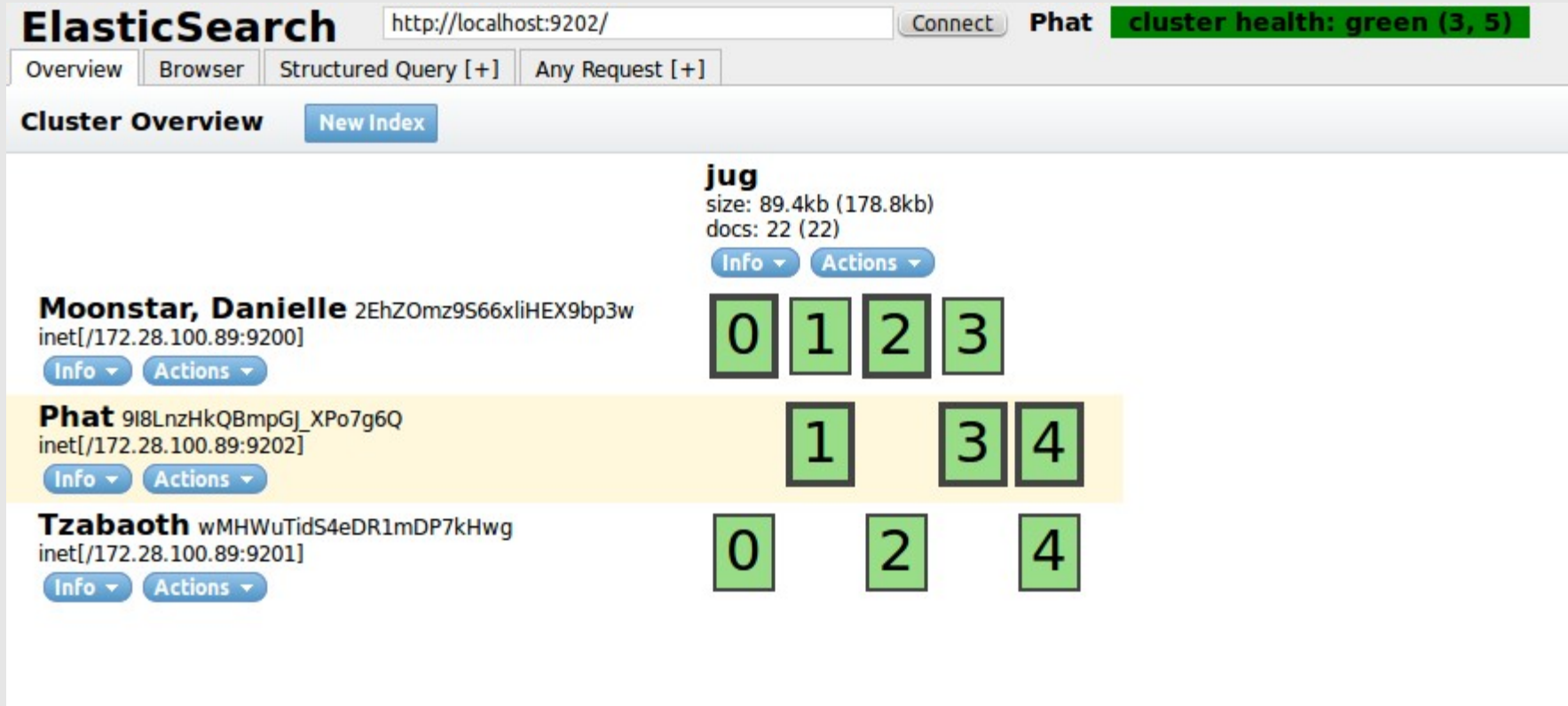
1

2

3

4

Verteilung





<http://lucene.apache.org>
<http://lucene.apache.org/solr/>
<http://elasticsearch.org>
<https://github.com/fhopf/lucene-solr-talk>

@fhopf
mail@florian-hopf.de
<http://blog.florian-hopf.de>

Images

- <http://www.morguefile.com/archive/display/3470>
- <http://www.morguefile.com/archive/display/695239>
- <http://www.morguefile.com/archive/display/93433>
- <http://www.morguefile.com/archive/display/811746>
- <http://www.morguefile.com/archive/display/12965>
- <http://www.morguefile.com/archive/display/181488>