CS 5604 Information Storage and Retrieval

Solr Team Final Presentation

Presenters:

Liuqing Li, Ye Wang, Anusha Pillai, Ke Tian

{liuqing, yewang16, anusha89, ketian} @vt.edu

Instructor:

Dr. Edward A. Fox

Virginia Polytechnic Institute and State University

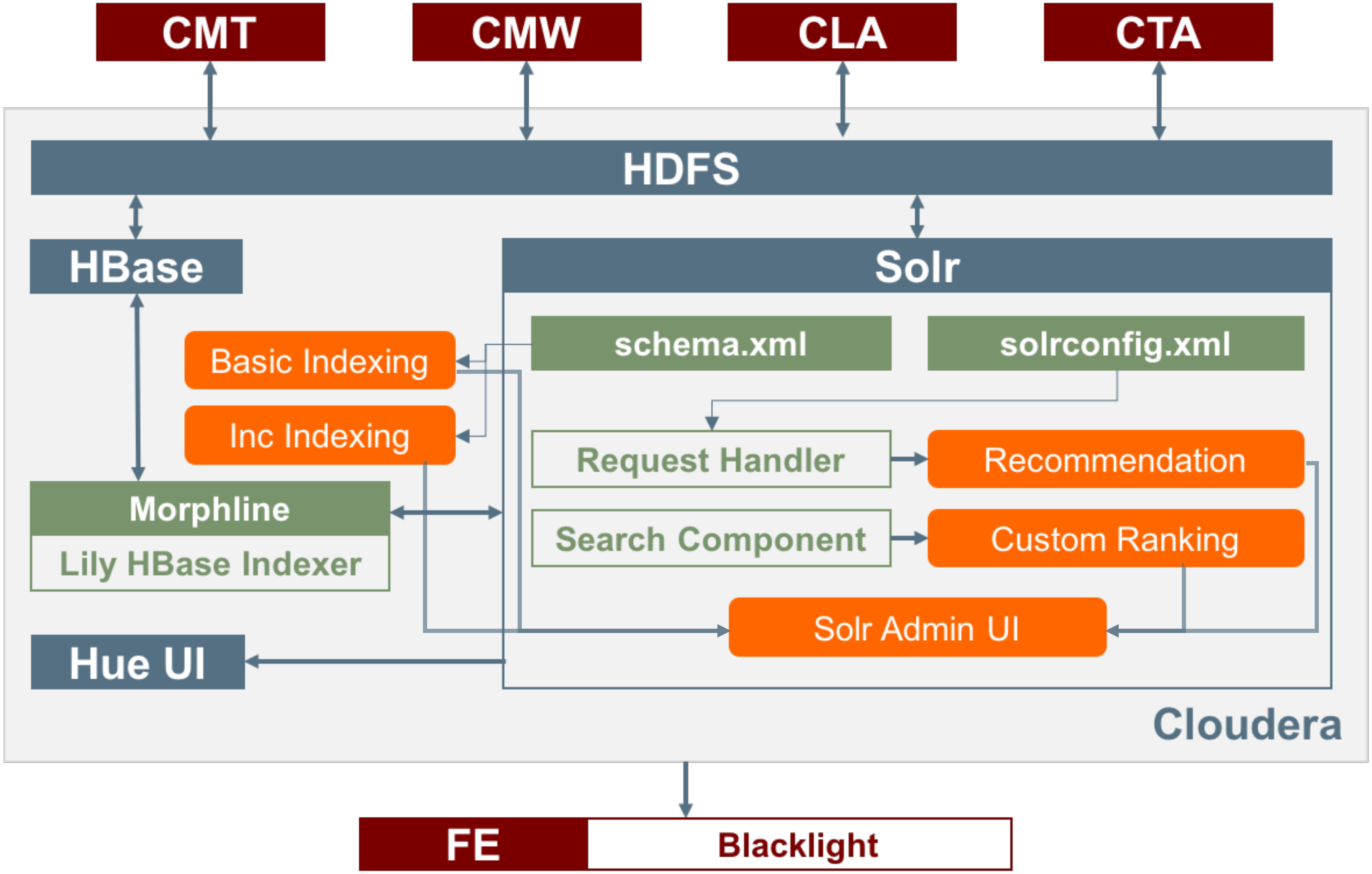
Blacksburg, VA, 24061

December 6, 2016

Outline

- Background
- Implementation
- Problems Faced
- Lessons Learned
- Future Work
- Acknowledgement

Background — Overview



Background — Updates

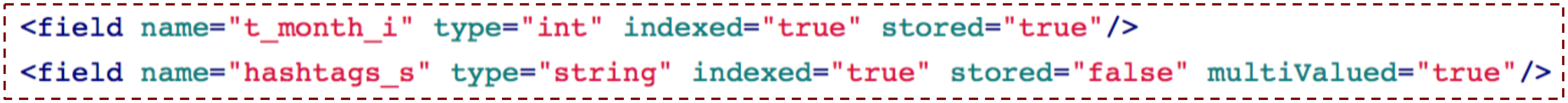
	Spring 2016	Fall 2016
schema.xml	Coarse grained	Fine grained
	No copyfields	Copyfields for all fields search
	Create stopwords.txt & profanity.txt	Update the two files
morphlines.conf	Two field types: string and text	Multiple field types
	Field “time” => string	Field “time” => datetime
	No multiple-valued fields	Multiple-valued field parser
Basic Indexing	Small collection	1.2 billion tweets dataset
Incremental Indexing	Virtual Cloudera (VC)	VC & Hadoop Cluster (HC)
Recommendation	Brief description	Implemented in VC & HC
Custom Ranking	Brief description	Implemented in VC & HC
Solr Admin UI	Brief description	Detailed description
	Limited faceted search	Detailed faceted search

Implementation — Basic Indexing

- Live Mode
  - Continuous stream of HBase cell updates into live search indexers
  - Simple and efficient
  - Cannot handle big data
- Batch Mode
  - Batch index tables in HBase by using MapReduce jobs
  - Write index files into HDFS (/user/cs5604f16_solr/...)
  - Can handle big data

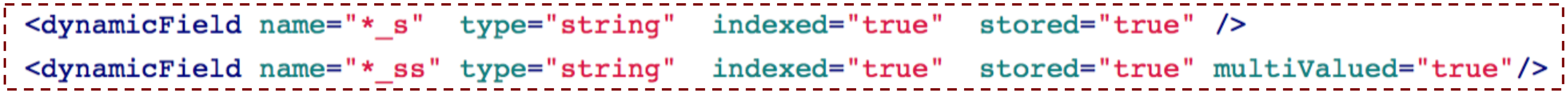
Implementation — Basic Indexing

- schema.xml: fields configuration
  - field (e.g., ideal-cs5604f16-fake)
    - # of fields: 30
    - Types: string (22), text_general (2), int (2), float (2), long (1), date (1)
    - Stored: True (17), False (13)



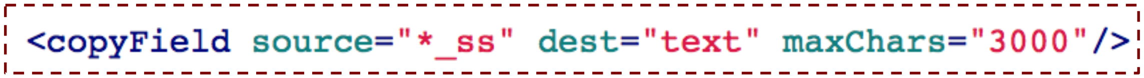
```
<field name="t_month_i" type="int" indexed="true" stored="true"/>
<field name="hashtags_s" type="string" indexed="true" stored="false" multiValued="true"/>
```

  - dynamicField: matching multiple fields, using wildcard



```
<dynamicField name="*_s"  type="string"  indexed="true"  stored="true" />
<dynamicField name="*_ss" type="string"  indexed="true"  stored="true" multiValued="true"/>
```

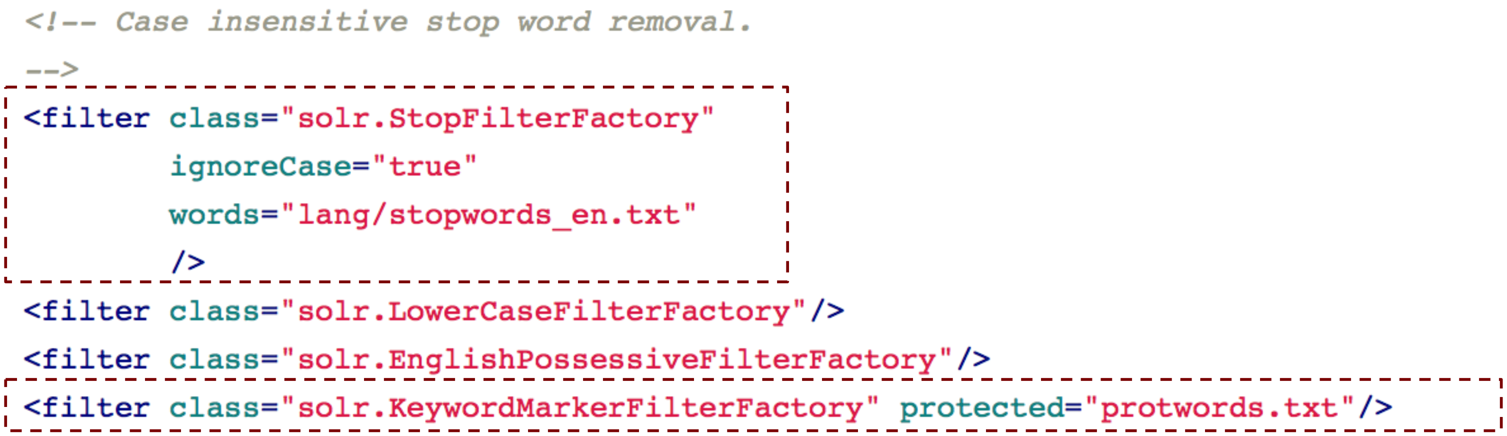
  - copyField



```
<copyField source="*_ss" dest="text" maxChars="3000"/>
```

Implementation — Basic Indexing

- stopword.txt and profanity.txt
  - stopword.txt: tf-idf value will not be calculated
  - profanity.txt: quick response for such search queries
  - Solr loads the two files while reading schema.xml



```
<!-- Case insensitive stop word removal.
-->
<filter class="solr.StopFilterFactory"
        ignoreCase="true"
        words="lang/stopwords_en.txt"
        />
<filter class="solr.LowerCaseFilterFactory"/>
<filter class="solr.EnglishPossessiveFilterFactory"/>
<filter class="solr.KeywordMarkerFilterFactory" protected="protwords.txt"/>
```

Source:
https://pypi.python.org/pypi/many-stop-words
http://www.freewebheaders.com/full-list-of-bad-words-banned-by-google/

Implementation — Basic Indexing

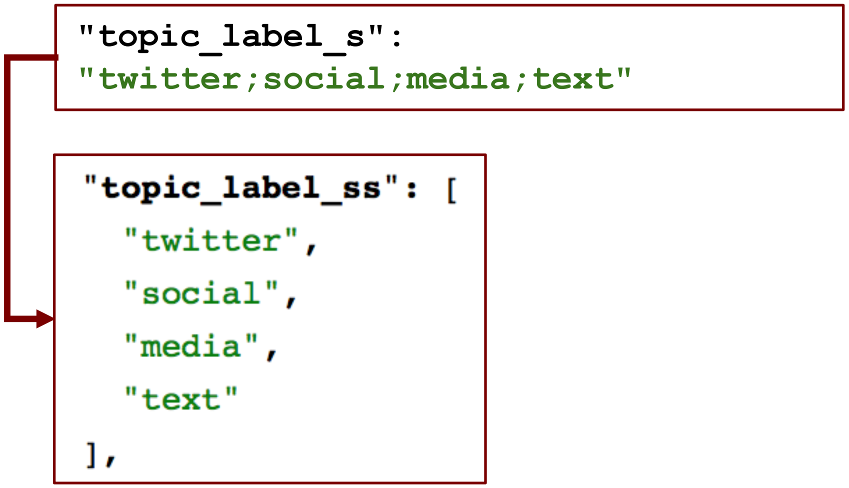
- morphlines.conf: mapping and parsing

Mapping data from HBase to Solr

```
mappings: [
# tweet : cleantext
{
        inputColumn: "tweet:cleantext"
        outputField: "raw_cleantext_s"
        type: string
        source: value
}
```

Split multiple values into list

```
split {
        inputField : "topic_label_s"
        outputField : "topic_label_ss"
        separator : ";"
        isRegex : false
        addEmptyStrings : false
        trim : true
}
```



```
"topic_label_s":
"twitter;social;media;text"
```

```
"topic_label_ss": [
  "twitter",
  "social",
  "media",
  "text"
],
```

Implementation — Basic Indexing

- Index the big dataset

		ideal-cs5604f16	ideal-cs5604f16-1204
Dataset		All collections (raw tweets)	All collections (raw tweets + processed data)
Indexing	# of DataNode	18	17
	Space Cost	392.33 GB	399.21 GB
	Time Cost		
	Mapping	1h21m	1h45m
	Reducing	5h11m	5h13m
	Merging	3h18m	3h10m
	Total	9h50m	10h8m

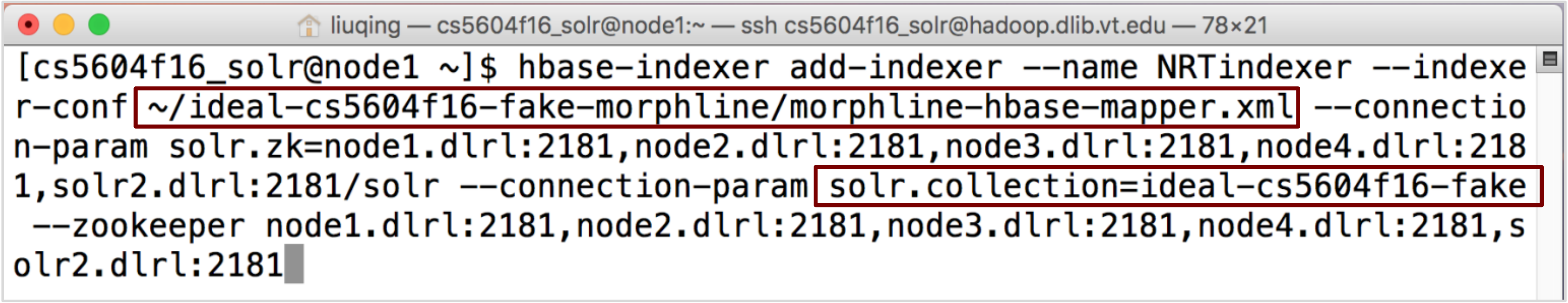
Implementation — Incremental Indexing

- Purpose
  - Process a continuous stream of HBase cell updates into live search indexes (Near Real-Time, NRT Indexing)
  - Solve the problem of frequent inserts, deletes and updates
- How does it work?
  - Enabling HBase replication (columnfamily)
  - Pointing an NRT Indexer Service at an HBase table
  - Starting an NRT Indexer Service
- Our work

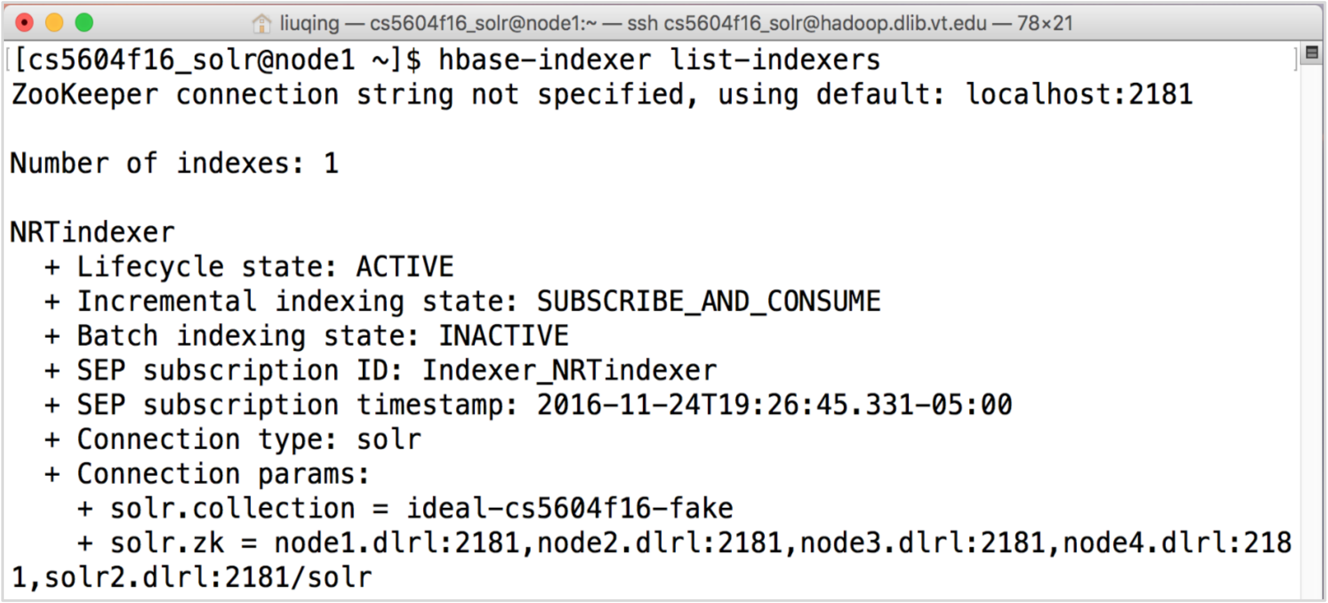
Source:
http://www.cloudera.com/documentation/enterprise/5-6-x/topics/search_config_hbase_indexer_for_search.html

Implementation — Incremental Indexing

Create and check the NRT indexer


liuqing — cs5604f16_solr@node1:~ — ssh cs5604f16_solr@hadoop.dlib.vt.edu — 78×21

```
[cs5604f16_solr@node1 ~]$ hbase-indexer add-indexer --name NRTindexer --indexe
r-conf ~/ideal-cs5604f16-fake-morphline/morphline-hbase-mapper.xml --connectio
n-param solr.zk=node1.dlrl:2181,node2.dlrl:2181,node3.dlrl:2181,node4.dlrl:218
1,solr2.dlrl:2181/solr --connection-param solr.collection=ideal-cs5604f16-fake
 --zookeeper node1.dlrl:2181,node2.dlrl:2181,node3.dlrl:2181,node4.dlrl:2181,s
olr2.dlrl:2181
```

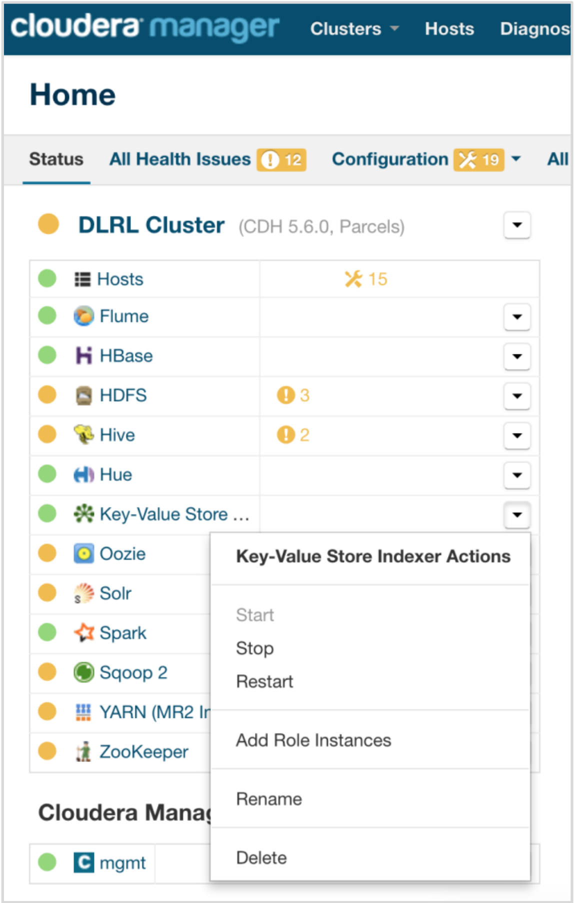

liuqing — cs5604f16_solr@node1:~ — ssh cs5604f16_solr@hadoop.dlib.vt.edu — 78×21

```
[cs5604f16_solr@node1 ~]$ hbase-indexer list-indexers
ZooKeeper connection string not specified, using default: localhost:2181

Number of indexes: 1

NRTindexer
  + Lifecycle state: ACTIVE
  + Incremental indexing state: SUBSCRIBE_AND_CONSUME
  + Batch indexing state: INACTIVE
  + SEP subscription ID: Indexer_NRTindexer
  + SEP subscription timestamp: 2016-11-24T19:26:45.331-05:00
  + Connection type: solr
  + Connection params:
    + solr.collection = ideal-cs5604f16-fake
    + solr.zk = node1.dlrl:2181,node2.dlrl:2181,node3.dlrl:2181,node4.dlrl:218
1,solr2.dlrl:2181/solr
```

Implementation — Incremental Indexing

Restart the HBase Solr Indexer service

Restart the service in HC



Home

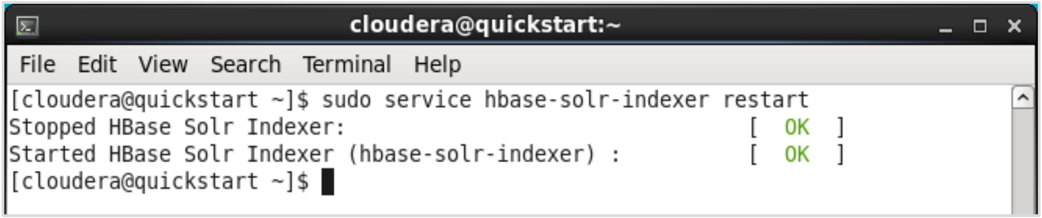
● DLRL Cluster (CDH 5.6.0, Parcels)

● Hosts 15
● Flume
● HBase
● HDFS 3
● Hive 2
● Hue
● Key-Value Store …
● Oozie
● Solr
● Spark
● Sqoop 2
● YARN (MR2 In
● ZooKeeper

Cloudera Manag

● mgmt

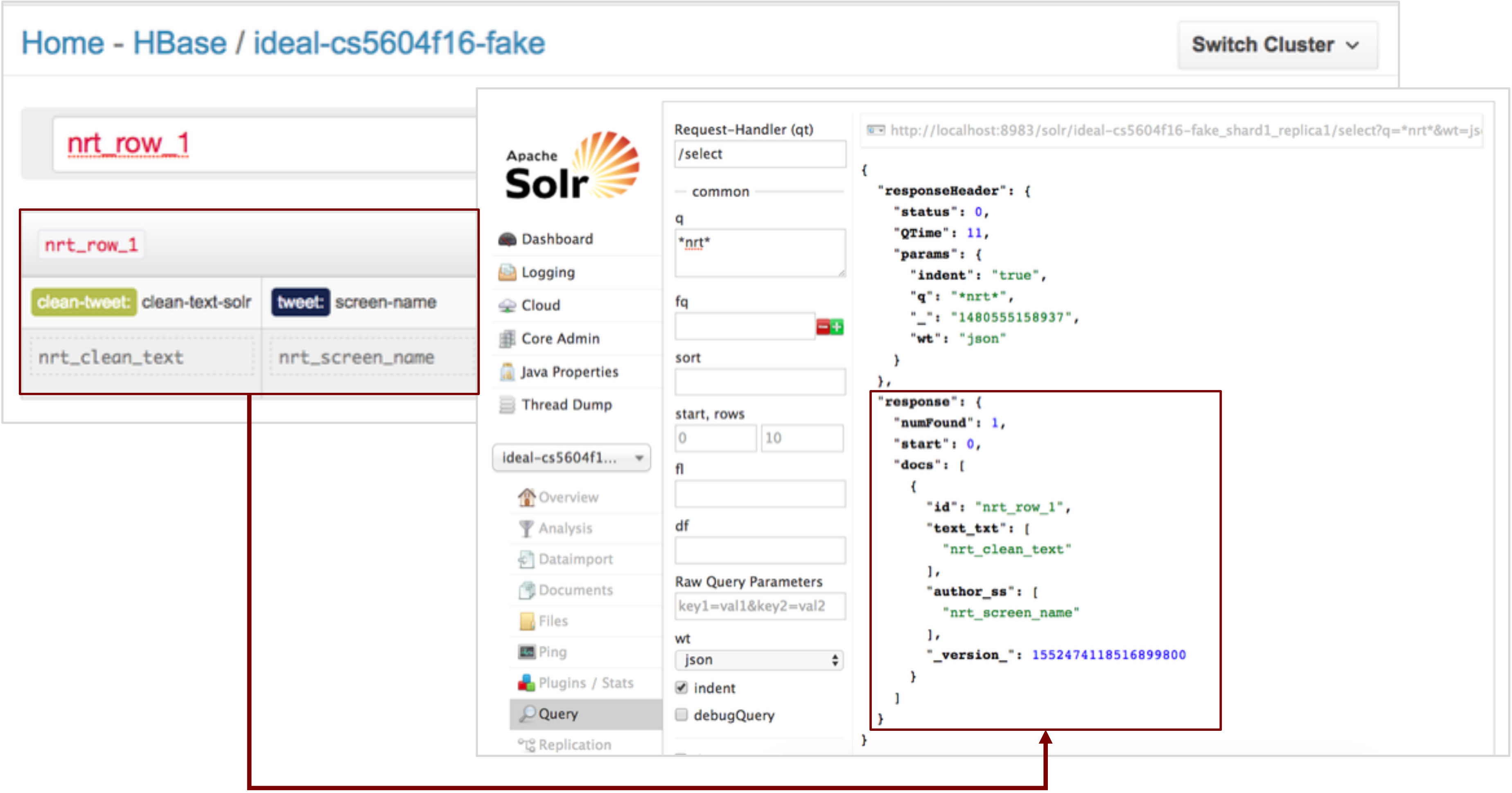
Restart the service in VC



```
[cloudera@quickstart ~]$ sudo service hbase-solr-indexer restart
Stopped HBase Solr Indexer:                                [  OK  ]
Started HBase Solr Indexer (hbase-solr-indexer) :          [  OK  ]
[cloudera@quickstart ~]$ 
```

Implementation — Incremental Indexing

Check the results in HBase and Solr Admin UI



Home - HBase / ideal-cs5604f16-fake

nrt_row_1

nrt_row_1	
clean-tweet: clean-text-solr	tweet: screen-name
nrt_clean_text	nrt_screen_name

Request-Handler (qt): /select

q: *nrt*

fq:

sort:

start, rows: 0 10

fl:

df:

Raw Query Parameters: key1=val1&key2=val2

wt: json

☑ indent

□ debugQuery

http://localhost:8983/solr/ideal-cs5604f16-fake_shard1_replica1/select?q=*nrt*&wt=js

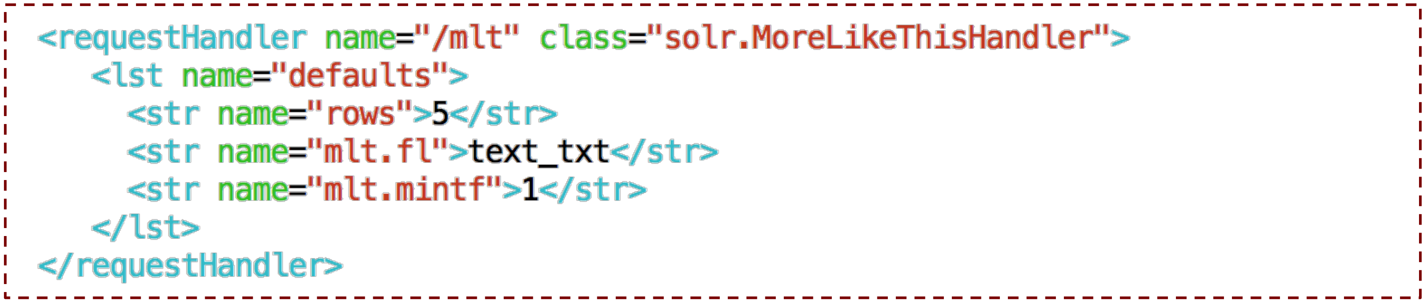
```
{
  "responseHeader": {
    "status": 0,
    "QTime": 11,
    "params": {
      "indent": "true",
      "q": "*nrt*",
      "_": "1480555158937",
      "wt": "json"
    }
  },
  "response": {
    "numFound": 1,
    "start": 0,
    "docs": [
      {
        "id": "nrt_row_1",
        "text_txt": [
          "nrt_clean_text"
        ],
        "author_ss": [
          "nrt_screen_name"
        ],
        "_version_": 1552474118516899800
      }
    ]
  }
}
```

Implementation — Recommendation

- Types
  - **Textual similarity based**
  - Collaborative filtering
- More Like This Component
  - Identifies similar documents to search result documents.
  - Can be configured as a **request handler** or search component
  - Uses term vectors to compute similarity.
  - Term vector can be calculated during query runtime or precomputed during indexing
  - Extracts highest matching terms based on tf-idf similarity

Implementation — Recommendation

- schema.xml
  - Set stored = true
  - Set termVectors = true (for calcalating tf-idf)
    - After making changes, reindexing is mandatory
- solrconfig.xml
  - Enable mlt

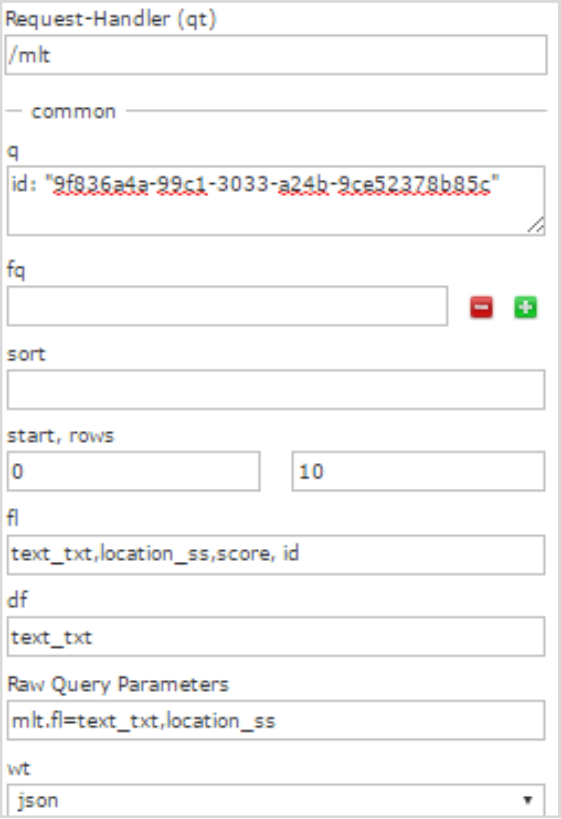


```
<requestHandler name="/mlt" class="solr.MoreLikeThisHandler">
   <lst name="defaults">
     <str name="rows">5</str>
     <str name="mlt.fl">text_txt</str>
     <str name="mlt.mintf">1</str>
   </lst>
</requestHandler>
```

  - Define other configuration parameters
  - e.g., mlt.fl, mlt.mintf, mlt.mindf, mlt.maxdf, mlt.qf

Implementation — Recommendation

- Request Handler



Request-Handler (qt)
/mlt

common

q
id: "9f836a4a-99c1-3033-a24b-9ce52378b85c"

fq

sort

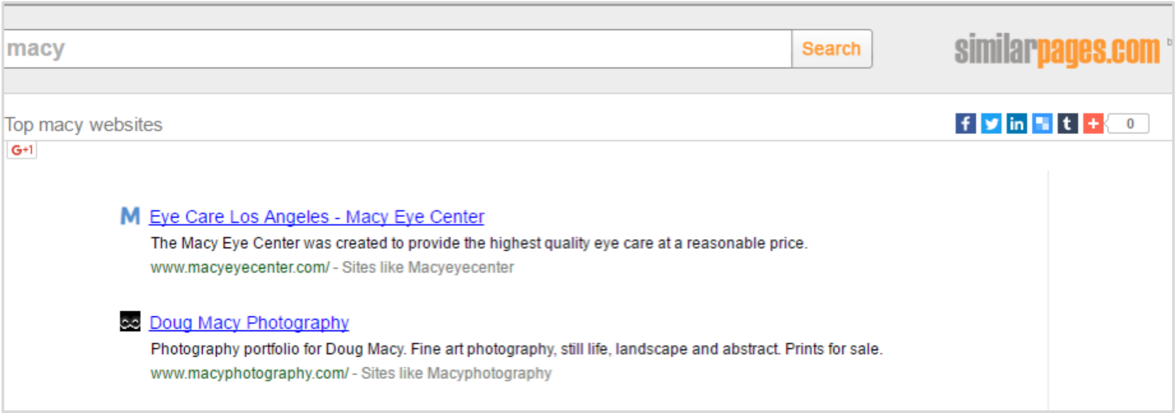
start, rows
0 10

fl
text_txt,location_ss,score, id

df
text_txt

Raw Query Parameters
mlt.fl=text_txt,location_ss

wt
json



macy

Top macy websites

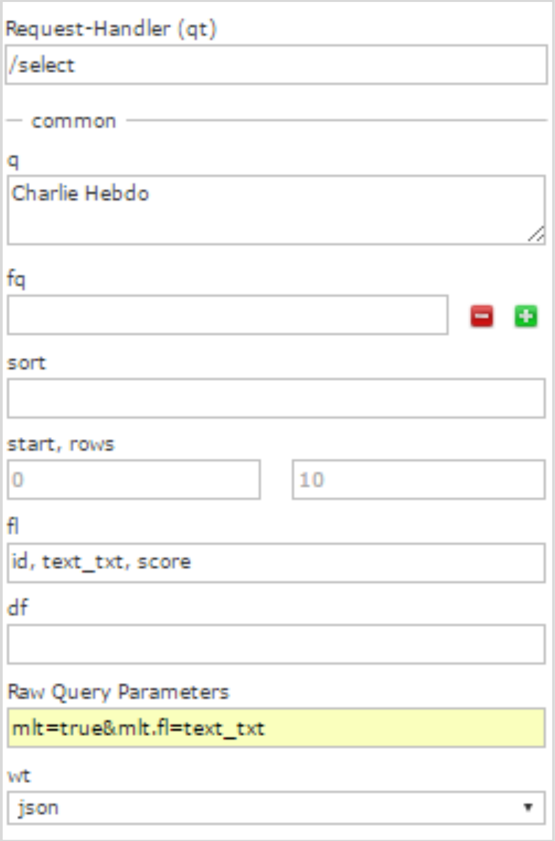
Eye Care Los Angeles - Macy Eye Center
The Macy Eye Center was created to provide the highest quality eye care at a reasonable price.
www.macyeyecenter.com/ - Sites like Macyeyecenter

Doug Macy Photography
Photography portfolio for Doug Macy. Fine art photography, still life, landscape and abstract. Prints for sale.
www.macyphotography.com/ - Sites like Macyphotography

Link:
https://drive.google.com/open?id=0B2iasHDgHqGyYUk0R3RkVktkM2M

Implementation — Recommendation

- Search Component



Request-Handler (qt)
/select

common

q
Charlie Hebdo

fq

sort

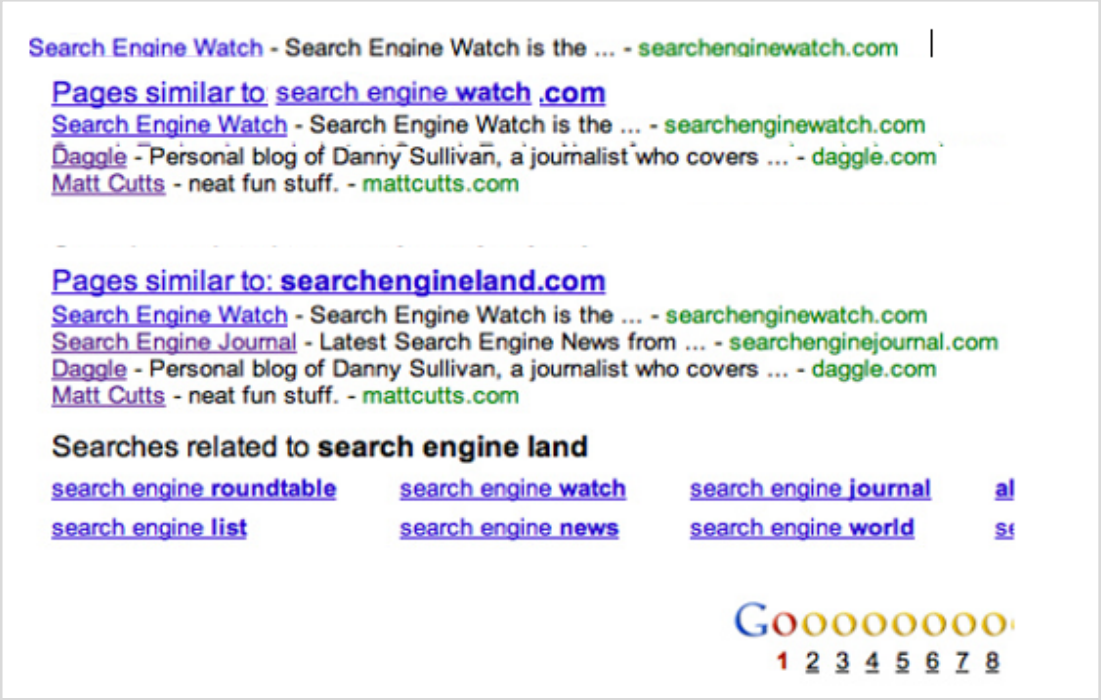
start, rows
0 10

fl
id, text_txt, score

df

Raw Query Parameters
mlt=true&mlt.fl=text_txt

wt
json



Search Engine Watch - Search Engine Watch is the ... - searchenginewatch.com

Pages similar to: search engine watch .com
Search Engine Watch - Search Engine Watch is the ... - searchenginewatch.com
Daggle - Personal blog of Danny Sullivan, a journalist who covers ... - daggle.com
Matt Cutts - neat fun stuff. - mattcutts.com

Pages similar to: searchengineland.com
Search Engine Watch - Search Engine Watch is the ... - searchenginewatch.com
Search Engine Journal - Latest Search Engine News from ... - searchenginejournal.com
Daggle - Personal blog of Danny Sullivan, a journalist who covers ... - daggle.com
Matt Cutts - neat fun stuff. - mattcutts.com

Searches related to **search engine land**

search engine roundtable | search engine watch | search engine journal | al
search engine list | search engine news | search engine world | se

Goooooooo
1 2 3 4 5 6 7 8

Link:
https://drive.google.com/open?id=0B2iasHDgHqGyU0doVEpidlh3c2c

Implementation — Custom Ranking

- Purpose
  - Customize and optimize the ranked results
- How does it work?
  - Search Component
    - prepare(): pre-processing, invoked before query is executed
    - processing(): post-processing, invoked after all the results are fetched
  - Custom Scoring

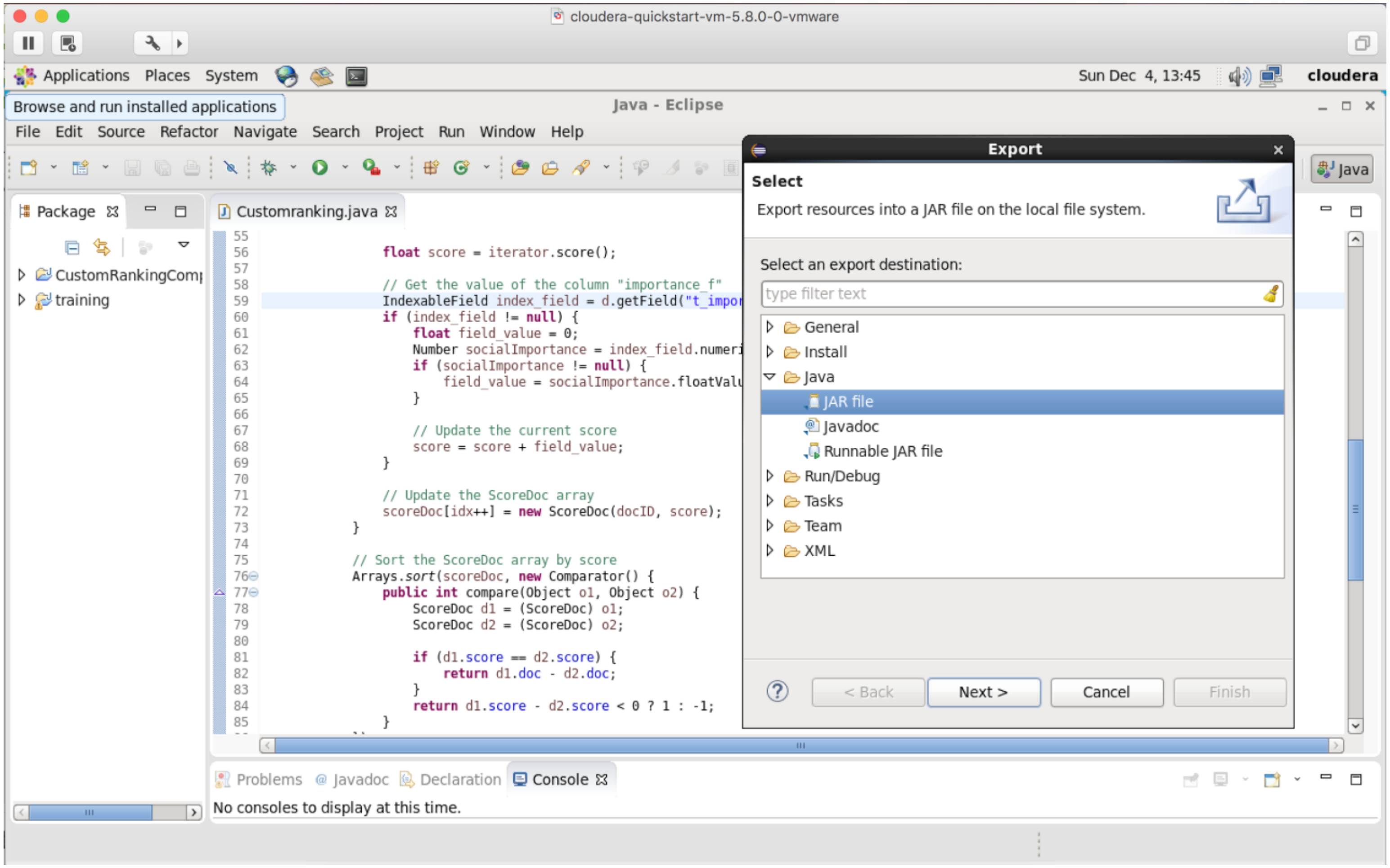
$$Score = Doc_{score,Solr} + Doc_{importance}$$

$$+ W_{topic} \times Doc_{score,topic} + W_{cluster} \times Doc_{score,cluster}$$

  - Re-ranking

Implementation — Custom Ranking

Build and copy jar file into Hadoop Cluster



```
                float score = iterator.score();

                // Get the value of the column "importance_f"
                IndexableField index_field = d.getField("t_impor
                if (index_field != null) {
                    float field_value = 0;
                    Number socialImportance = index_field.numeri
                    if (socialImportance != null) {
                        field_value = socialImportance.floatValu
                    }

                    // Update the current score
                    score = score + field_value;
                }

                // Update the ScoreDoc array
                scoreDoc[idx++] = new ScoreDoc(docID, score);
            }

            // Sort the ScoreDoc array by score
            Arrays.sort(scoreDoc, new Comparator() {
                public int compare(Object o1, Object o2) {
                    ScoreDoc d1 = (ScoreDoc) o1;
                    ScoreDoc d2 = (ScoreDoc) o2;

                    if (d1.score == d2.score) {
                        return d1.doc - d2.doc;
                    }
                    return d1.score - d2.score < 0 ? 1 : -1;
                }
```

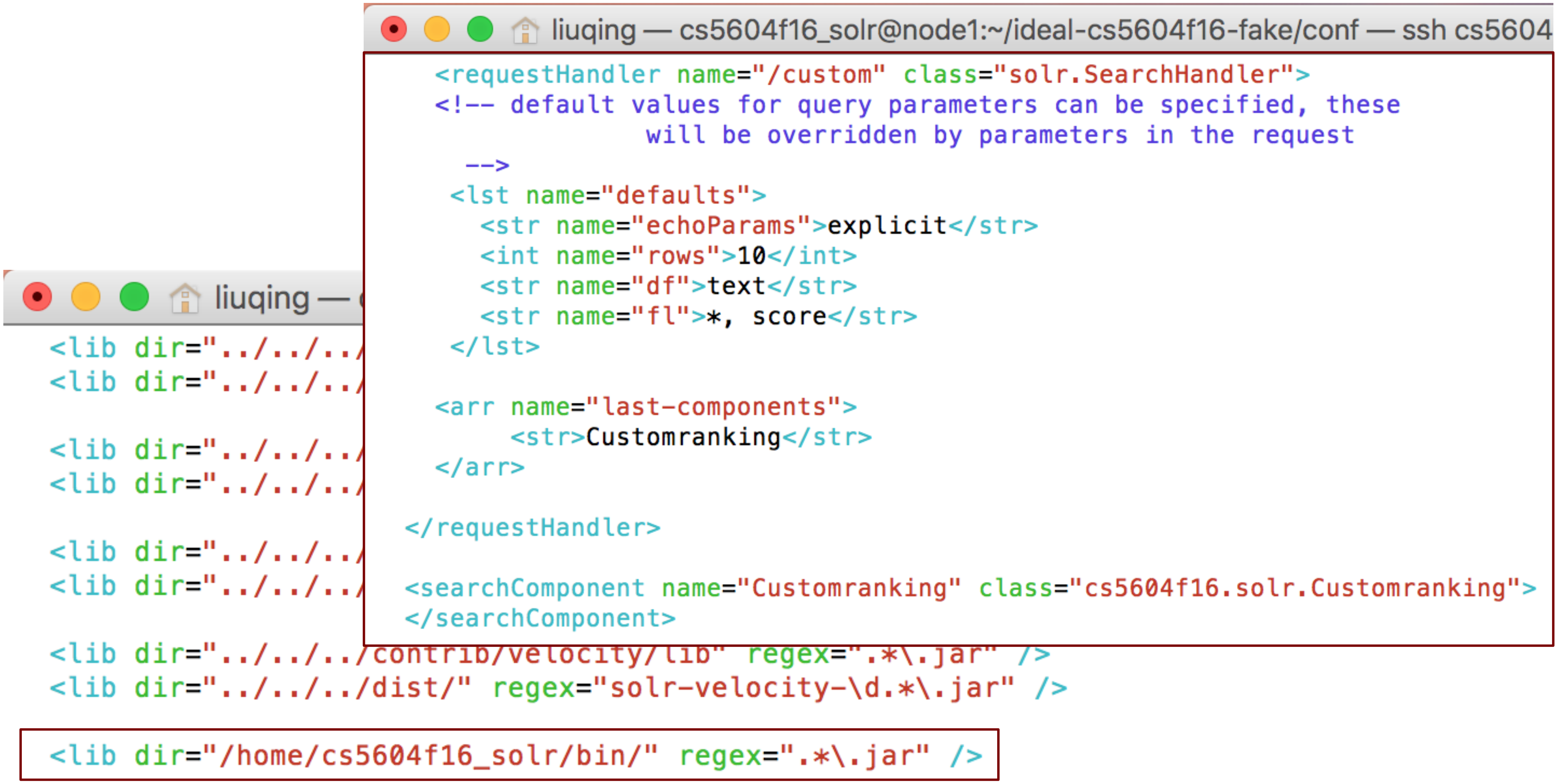
Export resources into a JAR file on the local file system.

Select an export destination:

No consoles to display at this time.

Implementation — Custom Ranking

Modify the solrconfig.xml



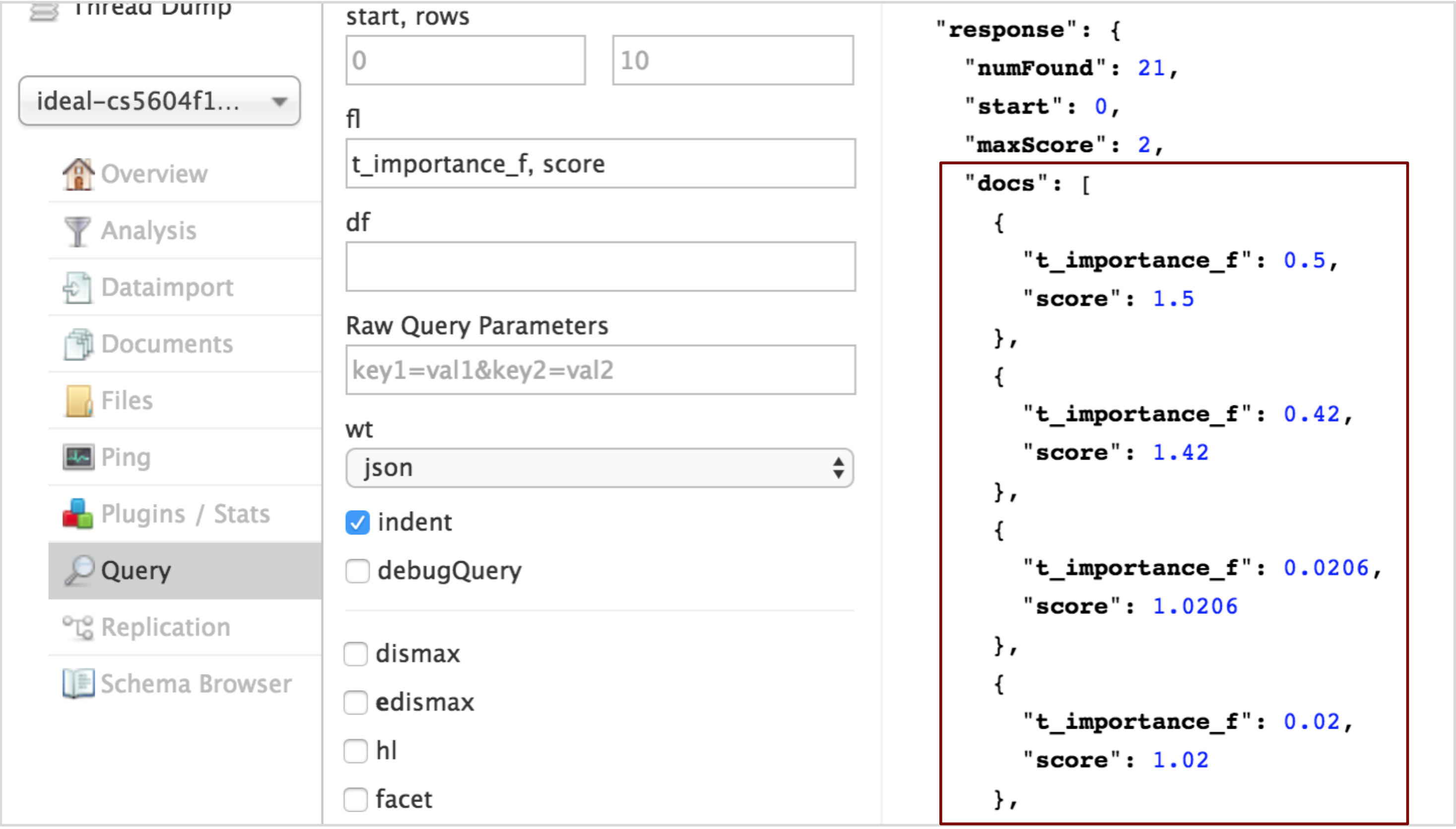
liuqing — cs5604f16_solr@node1:~/ideal-cs5604f16-fake/conf — ssh cs5604

```
   <requestHandler name="/custom" class="solr.SearchHandler">
   <!-- default values for query parameters can be specified, these
                 will be overridden by parameters in the request
     -->
    <lst name="defaults">
      <str name="echoParams">explicit</str>
      <int name="rows">10</int>
      <str name="df">text</str>
      <str name="fl">*, score</str>
    </lst>

   <arr name="last-components">
        <str>Customranking</str>
   </arr>

 </requestHandler>

 <searchComponent name="Customranking" class="cs5604f16.solr.Customranking">
 </searchComponent>
```

```
  <lib dir="../../../
  <lib dir="../../../

  <lib dir="../../../
  <lib dir="../../../

  <lib dir="../../../
  <lib dir="../../../

  <lib dir="../../../contrib/velocity/lib" regex=".*\.jar" />
  <lib dir="../../../dist/" regex="solr-velocity-\d.*\.jar" />

  <lib dir="/home/cs5604f16_solr/bin/" regex=".*\.jar" />
```

Implementation — Custom Ranking

- Update the instanceDir
- Reload the collection
- Check the results in Solr Admin UI



start, rows: 0 | 10

fl: t_importance_f, score

df:

Raw Query Parameters: key1=val1&key2=val2

wt: json

☑ indent

□ debugQuery

□ dismax

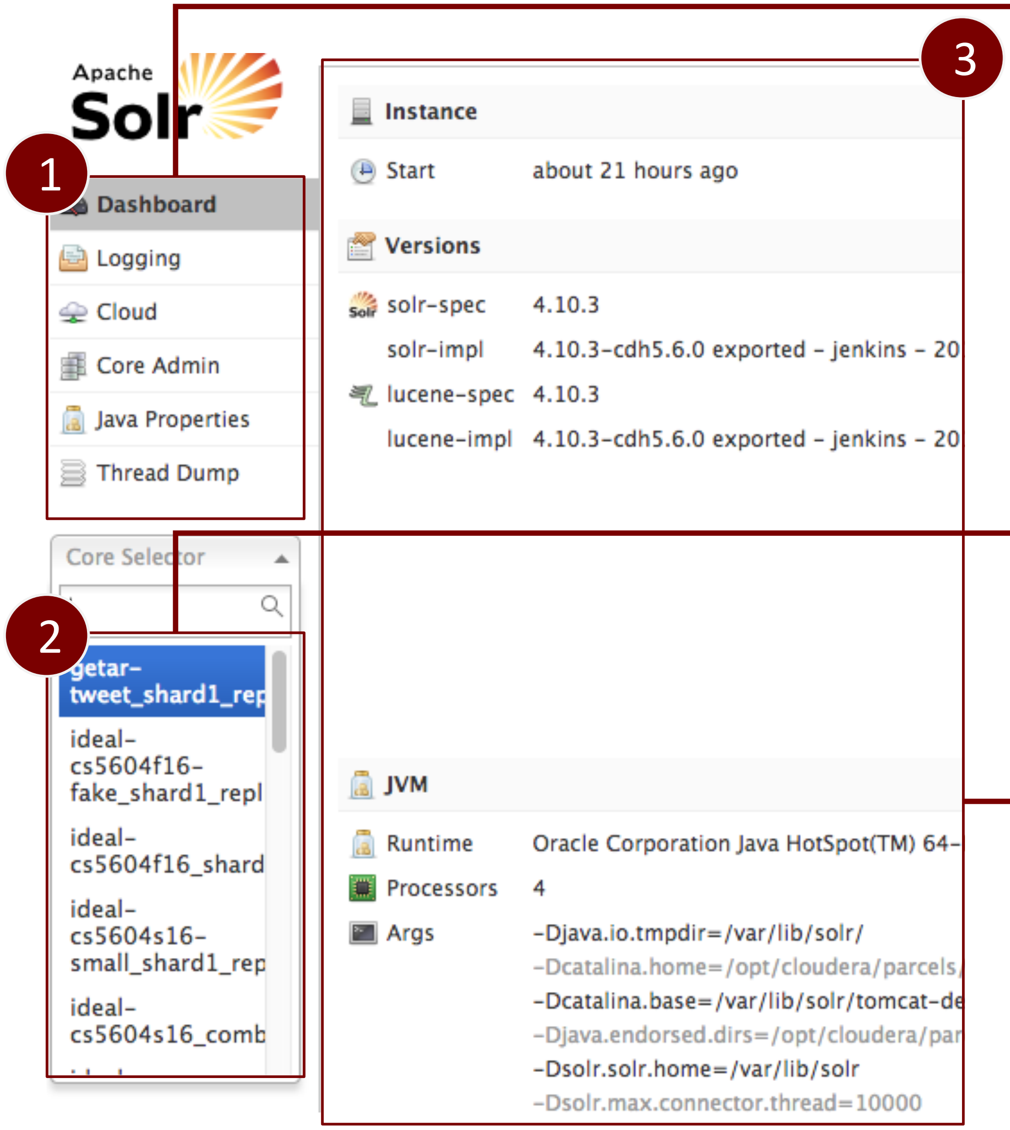
□ edismax

□ hl

□ facet

```
"response": {
  "numFound": 21,
  "start": 0,
  "maxScore": 2,
  "docs": [
    {
      "t_importance_f": 0.5,
      "score": 1.5
    },
    {
      "t_importance_f": 0.42,
      "score": 1.42
    },
    {
      "t_importance_f": 0.0206,
      "score": 1.0206
    },
    {
      "t_importance_f": 0.02,
      "score": 1.02
    },
```

Implementation — Solr Admin UI



Instance

Start about 21 hours ago

Versions

solr-spec 4.10.3

solr-impl 4.10.3-cdh5.6.0 exported - jenkins - 20

lucene-spec 4.10.3

lucene-impl 4.10.3-cdh5.6.0 exported - jenkins - 20

JVM

Runtime Oracle Corporation Java HotSpot(TM) 64-

Processors 4

Args
```
-Djava.io.tmpdir=/var/lib/solr/
-Dcatalina.home=/opt/cloudera/parcels/
-Dcatalina.base=/var/lib/solr/tomcat-de
-Djava.endorsed.dirs=/opt/cloudera/par
-Dsolr.solr.home=/var/lib/solr
-Dsolr.max.connector.thread=10000
```

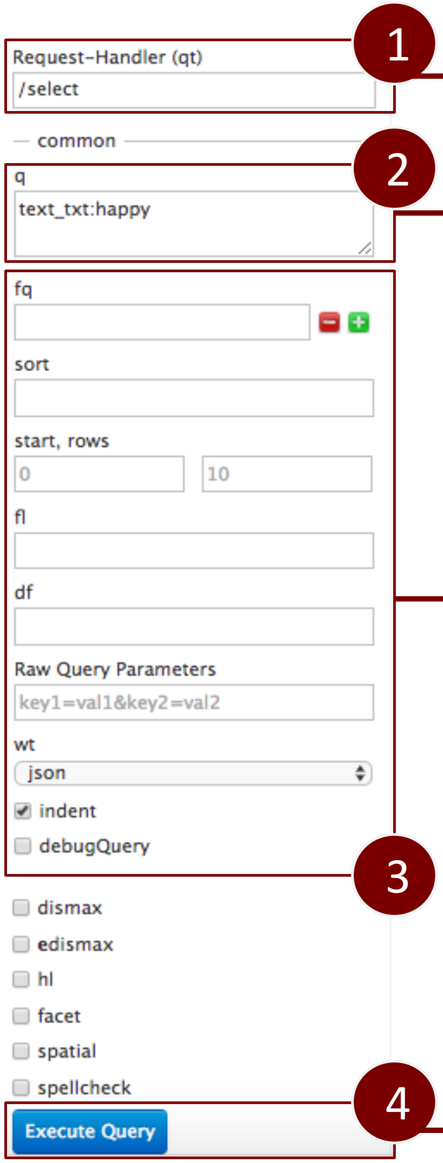
DashBoard: provide basic functions for users to choose. (Logging to check Solr logs for debugging)

Core Selector: select the core (dataset) for queries

Solr instance Information: current versions, JVM information

Choose ideal-cs5604f16-fake for querying

Implementation — Solr Admin UI



Request-Handler (qt)

/select

common

q

text_txt:happy

fq

sort

start, rows

0 10

fl

df

Raw Query Parameters

key1=val1&key2=val2

wt

json

☑ indent

□ debugQuery

□ dismax

□ edismax

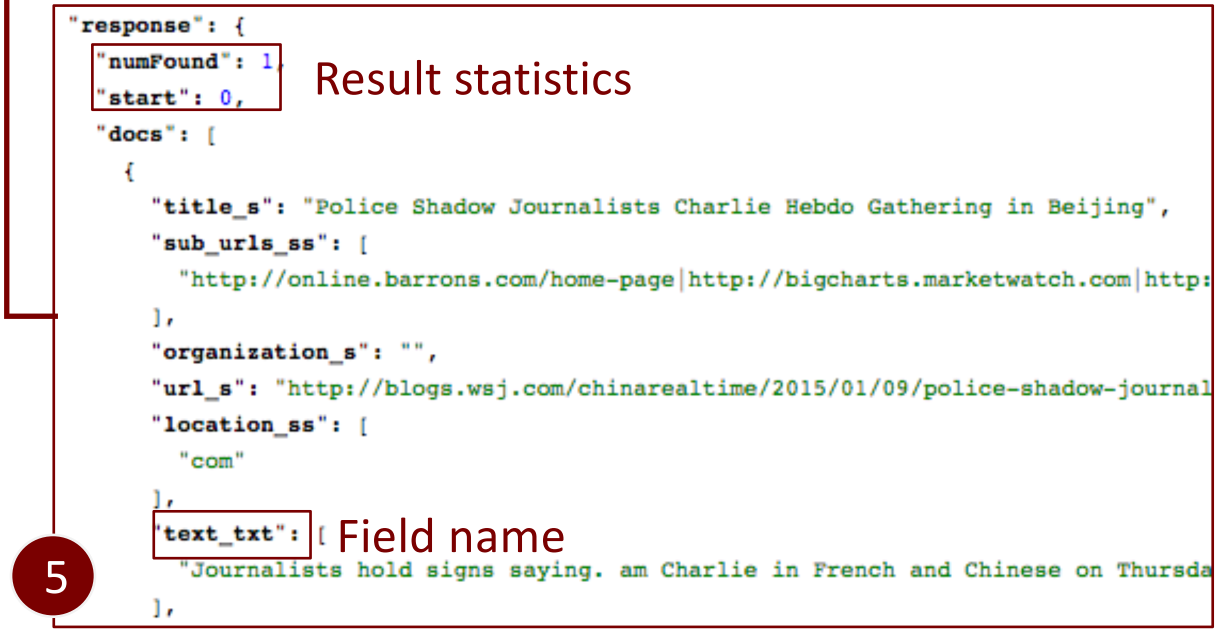
□ hl

□ facet

□ spatial

□ spellcheck

1. The request-handler: /select
2. The query event: q
3. Parameters for query: fq (filter queries) sort (descending or ascending)
4. Execute query
5. Results outputs: json format

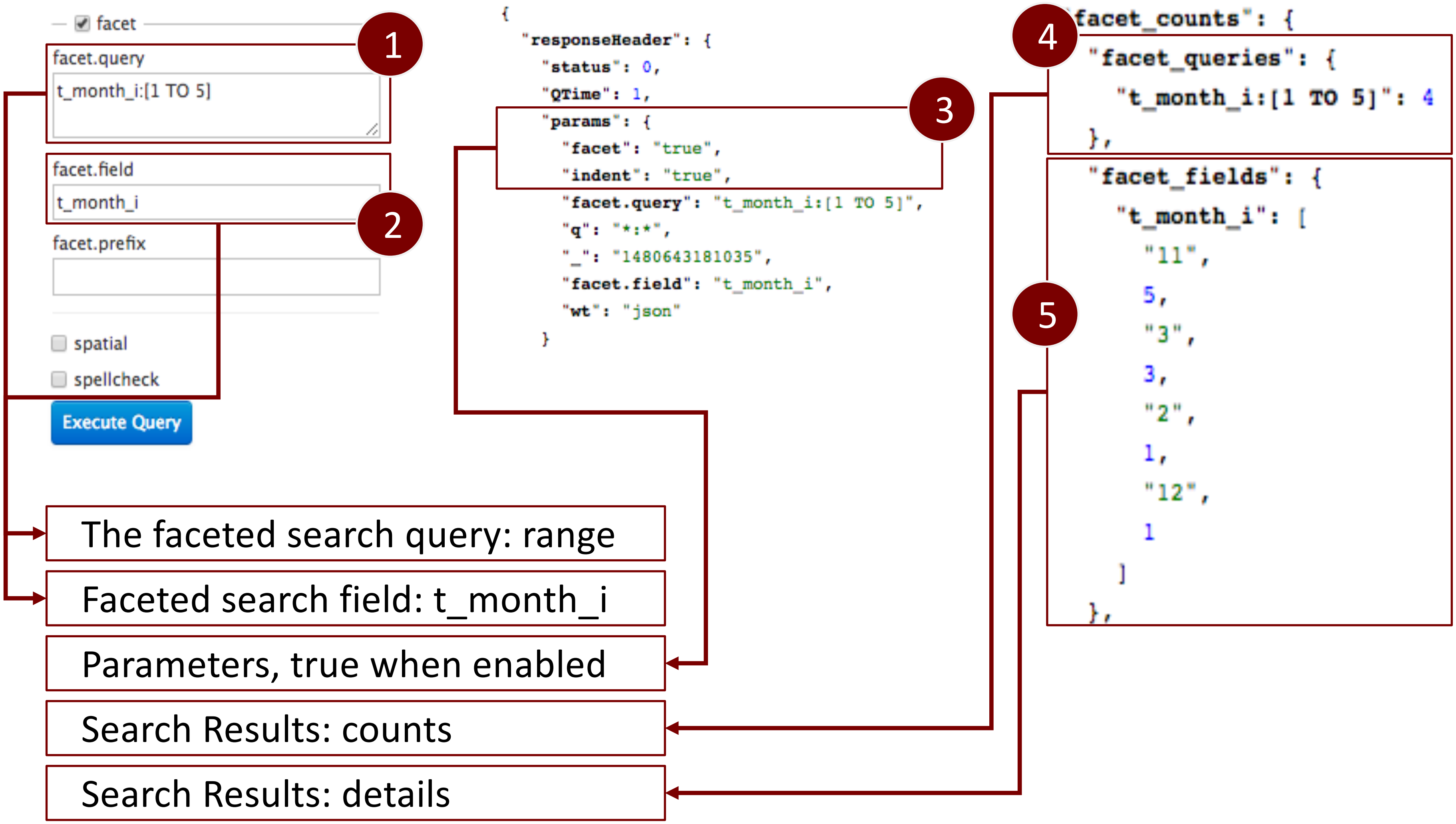


```
"response": {
  "numFound": 1,
  "start": 0,
  "docs": [
    {
      "title_s": "Police Shadow Journalists Charlie Hebdo Gathering in Beijing",
      "sub_urls_ss": [
        "http://online.barrons.com/home-page|http://bigcharts.marketwatch.com|http:
      ],
      "organization_s": "",
      "url_s": "http://blogs.wsj.com/chinarealtime/2015/01/09/police-shadow-journal
      "location_ss": [
        "com"
      ],
      "text_txt": [
        "Journalists hold signs saying. am Charlie in French and Chinese on Thursda
      ],
```

Result statistics

Field name

Implementation — Solr Admin UI



☑ facet

facet.query

t_month_i:[1 TO 5]

facet.field

t_month_i

facet.prefix

□ spatial

□ spellcheck

```
{
  "responseHeader": {
    "status": 0,
    "QTime": 1,
    "params": {
      "facet": "true",
      "indent": "true",
      "facet.query": "t_month_i:[1 TO 5]",
      "q": "*:*",
      "_": "1480643181035",
      "facet.field": "t_month_i",
      "wt": "json"
    }
```

```
"facet_counts": {
  "facet_queries": {
    "t_month_i:[1 TO 5]": 4
  },
  "facet_fields": {
    "t_month_i": [
      "11",
      5,
      "3",
      3,
      "2",
      1,
      "12",
      1
    ]
  },
```

The faceted search query: range

Faceted search field: t_month_i

Parameters, true when enabled

Search Results: counts

Search Results: details

Problem Faced

Cloudera and OS

- Virtual Cloudera seems slow and often crashes due to the memory
- Not familiar with the whole architecture at the beginning
- Versions of Cloudera and Solr

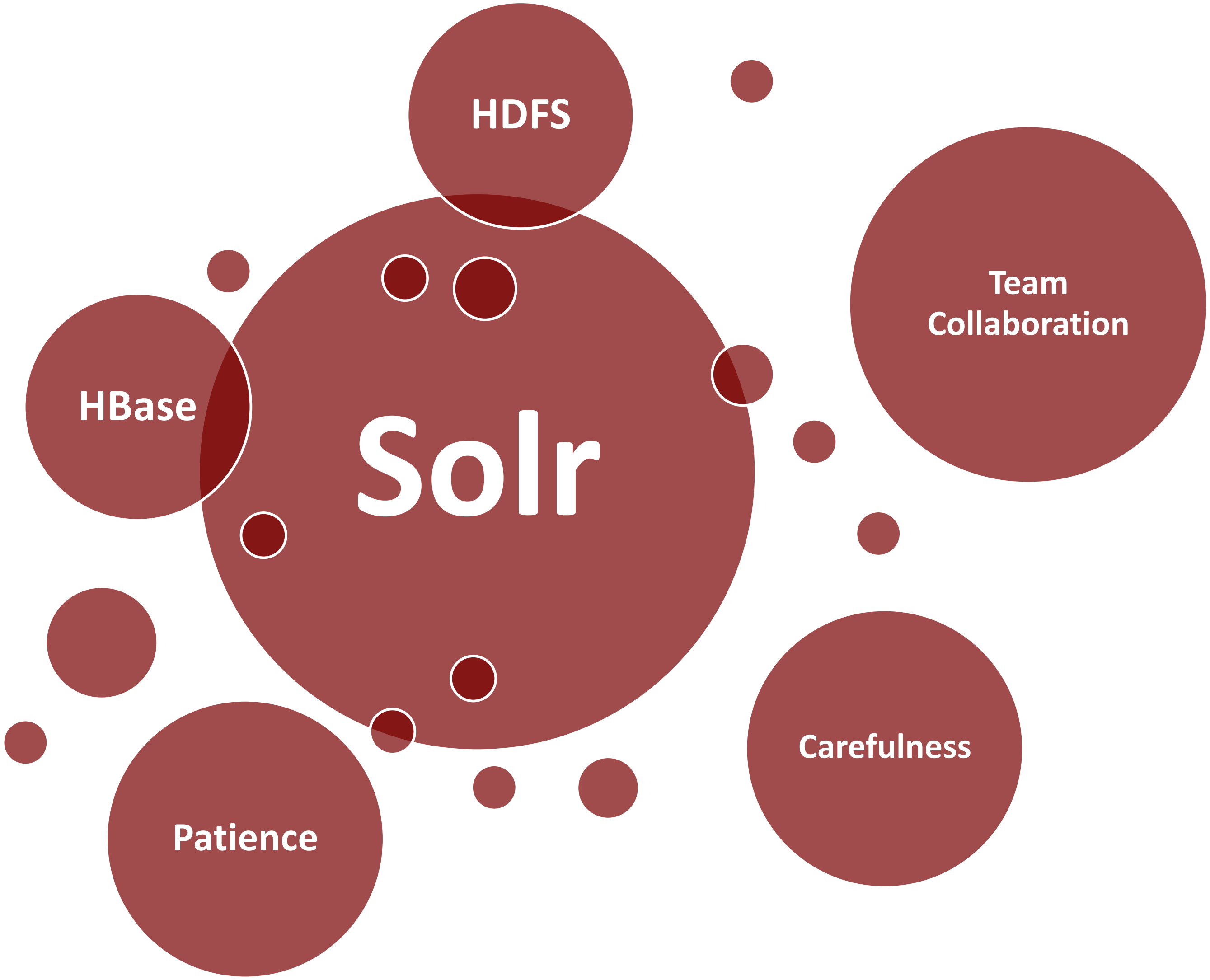
Data

- Consistency check
- Not enough real data available to perform tests
- Not much information available regarding logs to perform collaborative filtering

Collaboration

- Communication and modification

Lessons Learned



Future Work

Search

Customize more request handlers

Deal with the profanity issue

Custom Ranking

Customize more search components

Recommendation

Create a custom recommendation component (Probabilities – CTA team)

Implement the collaborative filtering (Log files – FE team)

Solr

Figure out SolrCloud, multiple Solr nodes in Cloudera Search

Acknowledgement

Projects	
NSF IIS - 1319578	III: Small: Integrated Digital Event Archiving and Library (IDEAL)
NSF IIS - 1619028	III: Small: Collaborative Research: Global Event and Trend Archive Research (GETAR)

Teams
CMT, CMW, CLA, CTA, FE teams

Persons	
Instructor	Dr. Edward A. Fox
GRA	Sunshin Lee

Thank you !

Questions?