

Full text search with open source Lucene

Marc Krellenstein
Chief Technology Officer, Elsevier
m.krellenstein@elsevier.com
April 24, 2007



Lucene/Solr overview

- What is it, what does it have
- What is it missing
- Where does it stands re open source
pro's and con's
- When should you use it

Lucene

- General purpose, open source full text search engine
- Developed late 90s by Doug Cutting
 - Ex-Xerox PARC, Apple, Excite
 - 3rd engine...(2nd = Excite)
 - Built to teach himself Java
 - Lucene is his wife's middle name
- Contributed to Apache project
 - <http://lucene.apache.org>

Current status

- 100% Java library, V2.0
- Free download, Apache license
- Ports to C++, .NET, Python, Perl
- Active development

Adoption

- Increased adoption last 3 years
- 1500+ installations:
 - HP, FedEx, Iron Mountain, Akamai, DSpace, IBM/Yahoo, Healthline, Webmail, CNET, Lookout (acquired by Microsoft), webshots.com (100M docs, 4M queries/day), Siderean, Monster....

Full-featured standard capabilities

- Field'd search w/multi-value support
- Boolean operators, +/- syntax
- Proximity operators, wildcards
- Run-time term weights
- Fuzzy search (edit distance)
- Term vectors (for find-similar, etc.)
- Range search
- Sort results by relevancy, date, field...
- Store documents and/or index them

Best practice relevancy ranking

- Term frequency
- Inverse document frequency
- Length normalization
- Proximity boosting
- Coordinate boost
- Run-time term weights
- Field weights
- Document weights
 - Boost by links, date, etc.

Best practice indexing

- New docs in separate partition for fast update
- Partition merging for performance
- Fast indexing (~= best competitors)
- Low disk overhead (<= best)
 - 15-20% of raw data for basic index
- Cross-platform index portability
 - Ex: build on Linux, search on Windows

Best practice query

- Excellent query speed and throughput
 - Sub-second query speed on large databases and high query volume
 - Comparable to and often better than competition
- Parallel search of multiple indexes
 - Could be local or remote

Scalability for very large databases

- 100M+ doc databases in production
- Can reportedly scale to 1B or more
 - 6B document test systems reported
 - But ...no systems that large in live production

Additional tools

- Index utility: Luke
- Open source converters
 - HTML, MS Office, PDF, XML, ...
- Open source ‘analyzers’ (tokenize, stem, multi-lingual support)
 - CJK, Romance languages, some others
- (Can license commercial versions
 - Multi-lingual support by Basis Tech)
- Solr...

Solr

- Enterprise search server based on Lucene available via HTTP
- Developed and contributed to open source by CNET
 - Incubated then released by Apache
- Thin layer...pure Lucene underneath

Solr capabilities

- Index schema and default field management
- Data loading scripts
- Caching and cache management
- Index replication
- Enhanced/package analyzers
- Web-based administration
- Faceted browse


File Edit View Favorites Tools Help

Back Search Favorites

Address <http://localhost:8080/solr/admin/>

Solr Admin (example)

ELSBURL-158238.home:8080
cwd=C:\jakarta-tomcat-5.0.28\solr-tomcat SolrHome=solr/



Solr [\[SCHEMA \]](#) [\[CONFIG \]](#) [\[ANALYSIS \]](#)
[\[STATISTICS \]](#) [\[INFO \]](#) [\[DISTRIBUTION \]](#) [\[PING \]](#) [\[LOGGING \]](#)

App server: [\[JAVA PROPERTIES \]](#) [\[THREAD DUMP \]](#)

Make a Query [\[FULL INTERFACE \]](#)

Query String:

Assistance [\[DOCUMENTATION \]](#) [\[ISSUE TRACKER \]](#) [\[SEND EMAIL \]](#)
[\[LUCENE QUERY SYNTAX \]](#)

Current Time: Mon Apr 23 21:35:17 EDT 2007
Server Start At: Mon Apr 23 21:33:27 EDT 2007

```
File Edit View Favorites Tools Help
Back Search Favorites
Address http://localhost:8080/solr/admin/get-file.jsp?file=solrconfig.xml
<?xml version="1.0" ?>
- <!--
Licensed to the Apache Software Foundation (ASF) under one or more
contributor license agreements. See the NOTICE file distributed with
this work for additional information regarding copyright ownership.
The ASF licenses this file to You under the Apache License, Version 2.0
(the "License"); you may not use this file except in compliance with
the License. You may obtain a copy of the License at

    http://www.apache.org/licenses/LICENSE-2.0

Unless required by applicable law or agreed to in writing, software
distributed under the License is distributed on an "AS IS" BASIS,
WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied.
See the License for the specific language governing permissions and
limitations under the License.
-->
- <config>
- <!--
Used to specify an alternate directory to hold all index data
other than the default ./data under the Solr home.
If replication is in use, this should match the replication configuration.
-->
- <!--
<dataDir>./solr/data</dataDir>
-->
- <indexDefaults>
  <!-- Values here affect all index writers and act as a default unless overridden. -->
  <useCompoundFile>false</useCompoundFile>
  <mergeFactor>10</mergeFactor>
  <maxBufferedDocs>1000</maxBufferedDocs>
  <maxMergeDocs>2147483647</maxMergeDocs>
  <maxFieldLength>10000</maxFieldLength>
  <writeLockTimeout>1000</writeLockTimeout>
  <commitLockTimeout>10000</commitLockTimeout>
</indexDefaults>
```

```
File Edit View Favorites Tools Help
Back Search Favorites
Address http://localhost:8080/solr/admin/get-file.jsp?file=solrconfig.xml
-->
<maxBooleanClauses>1024</maxBooleanClauses>
- <!--
  Cache used by SolrIndexSearcher for filters (DocSets),
  unordered sets of *all* documents that match a query.
  When a new searcher is opened, its caches may be prepopulated
  or "autowarmed" using data from caches in the old searcher.
  autowarmCount is the number of items to prepopulate. For LRUCache,
  the autowarmed items will be the most recently accessed items.
  Parameters:
  class - the SolrCache implementation (currently only LRUCache)
  size - the maximum number of entries in the cache
  initialSize - the initial capacity (number of entries) of
  the cache. (see java.util.HashMap)
  autowarmCount - the number of entries to prepopulate from
  and old cache.

-->
<filterCache class="solr.LRUCache" size="512" initialSize="512" autowarmCount="256" />
- <!--
  queryResultCache caches results of searches - ordered lists of
  document ids (DocList) based on a query, a sort, and the range
  of documents requested.

-->
<queryResultCache class="solr.LRUCache" size="512" initialSize="512" autowarmCount="256" />
- <!--
  documentCache caches Lucene Document objects (the stored fields for each document).
  Since Lucene internal document ids are transient, this cache will not be autowarmed.

-->
<documentCache class="solr.LRUCache" size="512" initialSize="512" autowarmCount="0" />
- <!--
  If true, stored fields that are not requested will be loaded lazily.

-->
<enableLazyFieldLoading>false</enableLazyFieldLoading>
- <!--
  Example of a generic cache. These caches may be accessed by name
  through SolrIndexSearcher.getCache(), cacheLookup(), and cacheInsert().
```

```
File Edit View Favorites Tools Help
Back Search Favorites
Address http://localhost:8080/solr/admin/get-file.jsp?file=solrconfig.xml
NOTE: there is "absolutely" nothing a client can do to prevent these
"invariants" values from being used, so don't use this mechanism
unless you are sure you always want it.

-->
- <lst name="invariants">
  <str name="facet.field">cat</str>
  <str name="facet.field">manu_exact</str>
  <str name="facet.query">price:[* TO 500]</str>
  <str name="facet.query">price:[500 TO *]</str>
</lst>
</requestHandler>
- <requestHandler name="instock" class="solr.DisMaxRequestHandler">
- <!--
  for legacy reasons, DisMaxRequestHandler will assume all init
  params are "defaults" if you don't explicitly specify any defaults.

-->
<str name="fq">inStock:true</str>
<str name="qf">text^0.5 features^1.0 name^1.2 sku^1.5 id^10.0 manu^1.1 cat^1.4</str>
<str name="mm">2<-1 5<-2 6<90%</str>
</requestHandler>
- <!--
queryResponseWriter plugins... query responses will be written using the
writer specified by the 'wt' request parameter matching the name of a registered
writer.
The "standard" writer is the default and will be used if 'wt' is not specified
in the request. XMLResponseWriter will be used if nothing is specified here.
The json, python, and ruby writers are also available by default.

<queryResponseWriter name="standard" class="org.apache.solr.request.XMLResponseWriter"/>
<queryResponseWriter name="json" class="org.apache.solr.request.JSONResponseWriter"/>
<queryResponseWriter name="python" class="org.apache.solr.request.PythonResponseWriter"/>
<queryResponseWriter name="ruby" class="org.apache.solr.request.RubyResponseWriter"/>

<queryResponseWriter name="custom" class="com.example.MyResponseWriter"/>

-->
```

File Edit View Favorites Tools Help

Back Search Favorites

Address <http://localhost:8080/solr/admin/stats.jsp>



SOLR Statistics (example)

ELSBURL-158238.home

Category	[CORE] [CACHE] [QUERY] [UPDATE] [OTHER]
	Current Time: Mon Apr 23 21:41:43 EDT 2007
	Server Start Time: Mon Apr 23 21:33:27 EDT 2007

CORE

name: Searcher@bc5596 main
class: org.apache.solr.search.SolrIndexSearcher
version: 1.0
description: index searcher
stats: caching : true
numDocs : 30571
maxDoc : 30633
readerImpl : MultiReader
readerDir : org.apache.lucene.store.FSDirectory@C:\jakarta-tomcat-5.0.28\solr-tomcat\solr\data\index
indexVersion : 1171129802817
openedAt : Mon Apr 23 21:33:40 EDT 2007
registeredAt : Mon Apr 23 21:33:40 EDT 2007

QUERY HANDLERS

name: standard
class: org.apache.solr.request.StandardRequestHandler
version: 1.0
description: The standard Solr request handler

What is it missing

- Proven production scalability at 1B+ docs
- Sophisticated packaging, documentation, configuration
- Full set of doc converters, languages
- Security
- User interface
- Non-core features: Alerts, text mining, dynamic summaries,...
- Commercial support, services

Open source benefits in general

- Free...including re-distribution?
- Control: customize as needed
 - Only realistic if source code is good
- Open source, community support:
 - Application focus
 - But only valuable if active community
- No make vs. buy (already made)
- No hype
- (Participation in open source?)

Open source risks in general

- No commercial support, professional services
 - Must assure support
 - ‘No one to choke’
- Possible legal exposure if product origins are controversial
- Possible product weaknesses in
 - Packaging – install, configuration, ...
 - Features (that programmers don't like)
 - Admin tools (that programmers don't like)
 - Robustness or scalability

Open source benefits as re Lucene

- All uses are free, including re-distribution
 - Apache only requires copyright notice
- Good ability to customize: 100%
Java, beautifully written
- Community support: Many committers, active forums, email list
- (No hype)

Open source risks as re Lucene

- No commercial support
 - Lucene consultants available
 - No substantial 3rd party support
 - Need to support with internal staff
 - Could outsource...give them the source
 - (Community support not adequate for mission-critical applications)
- Legal exposure: none...clear provenance
- Product weaknesses...

Product weaknesses

- Packaging
 - Solr helps
 - Still lacking re assembling pieces, doc
- Features
 - No non-core: alerts, text mining, ...
- Admin tools – adequate but limited
- Robustness or scalability
 - Not proven in production \geq 1B docs

Product summary

- High quality, high performing search engine commodity
- Full featured (but not every feature)
- Scalable
- Highly customizable
- Free
- Requires sophistication re support, setup

Indicators for use (some or all...)

- Ability to provide adequate support
- Sufficient tech sophistication to deal with limited packaging, configuration
- Need for high throughput, query speed, scale
- Need for good search precision
- Need to minimize cost
- Need for low disk overhead
- Need for control or customization