Using Apache Solr for Ecommerce Search Applications

Rajani Maski
Happiest Minds, IT Services

SHARING. MINDFUL. INTEGRITY. LEARNING. EXCELLENCE. SOCIAL RESPONSIBILITY.

Copyright Information

This document is exclusive property of Happiest Minds Technologies Pvt. Ltd. It is intended for limited circulation.

Contents

Copyright Information 2
Abstract 4
Apache Solr Overview 4
Advantages of Apache Solr 5
Apache Solr: Architecture 5
Case Study: Ecommerce Search Application 6
Apache Solr's features detailed 8
Sample Search Site 10
Conclusion 11
References 12

happiest minds

© 2013 Happiest Minds Technologies Pvt. Ltd. All Rights Reserved

Abstract

Online retailers are finding unprecedented growth in sales over the past few years. Forrester statistics predicts that online sales growth in the U.S. shall trend at a 10% compound annual growth rate through 2014[1]. Ecommerce, which generated $231 billion in sales for U.S. retailers last year, is expected to increase by 13% i.e., $262 billion this year. The vital reason for such incremental growth is the invention of faster internet connectivity and powerful online tools resulting in a new commerce arena. However, building a smart business website requires lot of consideration. The ability for shoppers to locate products within an online store quickly and easily is of the utmost importance. According to Forrester statistics, around 60% of online purchases result from a customer search and half of online shoppers use the site search box while shopping. Therefore, business needs to ensure that the website's search tools are simple and intuitive.

An effective search function for an e-commerce site has a number of potential benefits. Firstly, search entitles the customers to shop easily with navigations and filters. Secondly, it also allows customer to identify, qualify and compare a product with other products based on specifications and other customers' feedback. Finally, it enables the customer to enter his search query in a plain search text box and lets the search engine to find products for him. Customers are accustomed to obtain accurate and quick results from search engines, and will expect a similar experience on e-commerce sites. Therefore, an agile search engine with prospective search functionalities is essential for the increase in the online sales. In this paper we are going to present how Apache Solr can be used as the search engine in an ecommerce platform.

Apache Solr Overview

Apache Solr is a search platform focused on delivering enterprise class, high performance search functionality. It can be used to power search features within any data driven websites. Solr can be integrated with any application or website because it communicates using standard formats such as HTTP, XML and JSON. Solr's REST-like APIs make it easy to use from virtually any programming language. Its major features are full-Text search capabilities, faceted search, spatial search, and distributed search. These features are of great significance because search is not just about returning result-set objects that match words in a user query. It includes processing of the original text into indexed terms followed by relevancy ranking and returning results in special grouping formats such as facets, clusters and many more.

Also, Solr has the capability of distributed searching, fault-tolerance and load balancing and therefore it is extensively used in a big-data ecosystem.

happiest minds

© 2013 Happiest Minds Technologies Pvt. Ltd. All Rights Reserved

Advantages of Apache Solr

With the increase in number of online users, application scalability and reliability is of prime importance. Solr is proven in terms of reliability, scalability, and speed; it is such an engine with which inbound traffic can be increased, through ensuring that search results and ad-hoc combinations of search terms are most desirably exposed for public Web searching.

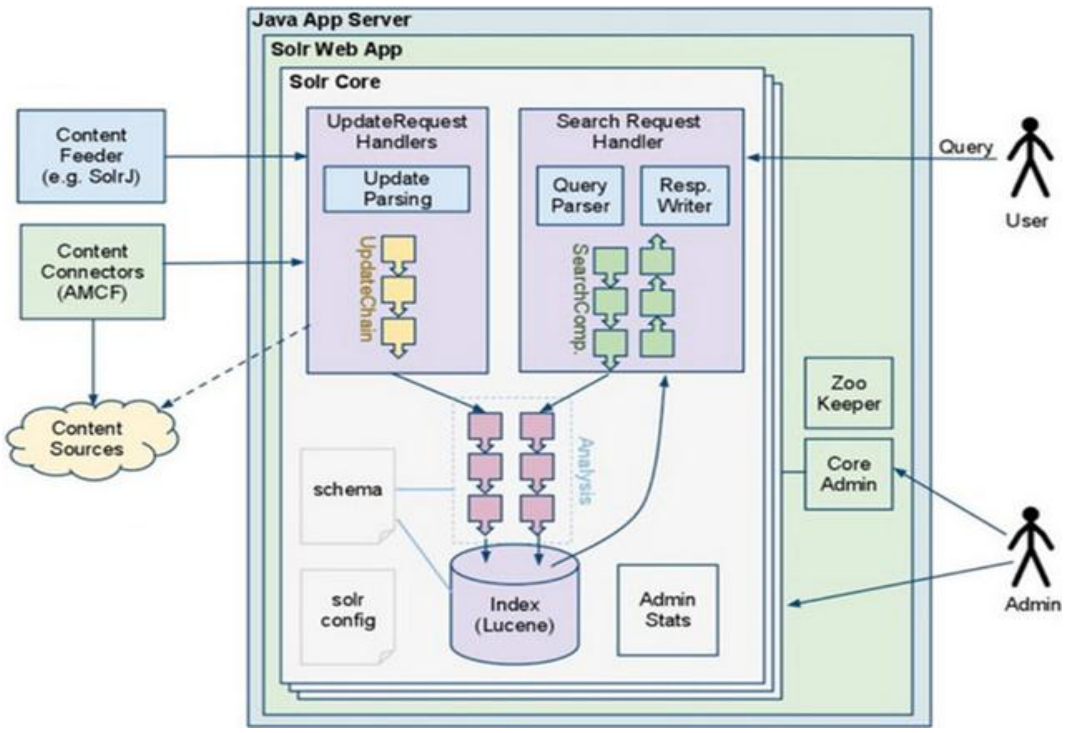
Apache Solr now supports several types of joins and grouping options that can be extensively used when there is a need of normalization to preserves documents relationship. It has an optional column-oriented storage called *docValues* that is the way to build the forward index. And out of box, it gives the developer, the flexibility to have their own complex data types and storage, ranking, and analytics functions.

The Apache Lucene's libraries that Apache Solr has in its core are widely used in most of the scalable websites such as Twitter, LinkedIn and other social networking sites, etc. Such websites are estimated to manage up-to 12000 queries per second and with Indexing speed of 220GB/hour for 4k documents.

Ease of use, relevance, find-ability and recommending other results are always the important aspects of search technology. Apache Solr with its text analyzing capabilities such as Automatic term stemming, spell-correction, synonym search, multi-lingual analysis and etc., enable customers to enter free text search and retrieve products of their interest. Finally, it is the most widely deployed search platform, and is supported and developed by a large community of open source developers and committers.

Apache Solr: Architecture

[Figure 1 Image ref link [1]]



© 2013 Happiest Minds Technologies Pvt. Ltd. All Rights Reserved

To understand the pluggable capabilities, let's have a brief understanding on each component involved in a Solr search.

1. When a user runs a search in Solr, the search query is processed by a request handler. As show in the above diagram (Figure 1), on the right hand side, the *search request handler* is a plugin that defines the logic to be used when Solr processes a request. Solr supports a variety of request handlers. In addition, applications can be configured to allow users to override the default handlers in preference of a custom request handler.
2. The search request handler includes query parser and response writer. To process a search query, a request handler calls a *query parser* that is responsible for parsing the textual query and converting it into corresponding Lucene query objects. Different query parsers support different syntax. Solr, by default, includes a standard (earlier Lucene) query parser for greater precision in searches.
3. The *response writer* component builds the query response object in the required format for the final presentation, generally an XML/JSON object is returned.
4. Solr has the cloud architecture that enables the distributed capabilities. It's a system in which data is organized into multiple instances, or shards, that can be hosted on multiple machines, with replicas providing redundancy for both scalability and fault tolerance, and a ZooKeeper server that helps manage the overall structure so that both indexing and search requests routed are in sync.

Case Study: Ecommerce Search Application

The foremost requirement of an e-commerce search platform is the search text box on each page of the site that allows user to enter their search query and tell system what exactly they are looking for. System should be smart to do the language processing and retrieve high quality relevant products. The search results should be focused on displaying the product information excluding other general content of website. Also system should be able to recommend products based on what user is looking for and also display other user's feedback for that product. Other common features that a user commonly looks for are sort based on price, popularity and relevance, auto-suggestions, synonymous results, spell correction & etc. Figure 3 is the glance of a sample search site with all the above features embedded. System should also be able to do the language analysis and accept frequent incremental updates such as stock availability and clicks.

However, Apart from the customary e-commerce features listed above, relevant recommendation engine that finds the most interesting products for a customer from the complete catalog plays a crucial role in driving a positive user experience and building customer loyalty. Recommendation engines have become an essential tool for online vendors, retailers and others. Apache Solr integrated with Hadoop provides a robust, efficient, and flexible platform to build such smart recommending engines. It also gives an integral system that captures customers' behavior from multiple data sources and hence enabling to drive new business opportunities and features from it.

happiest minds

© 2013 Happiest Minds Technologies Pvt. Ltd. All Rights Reserved

Solution overview

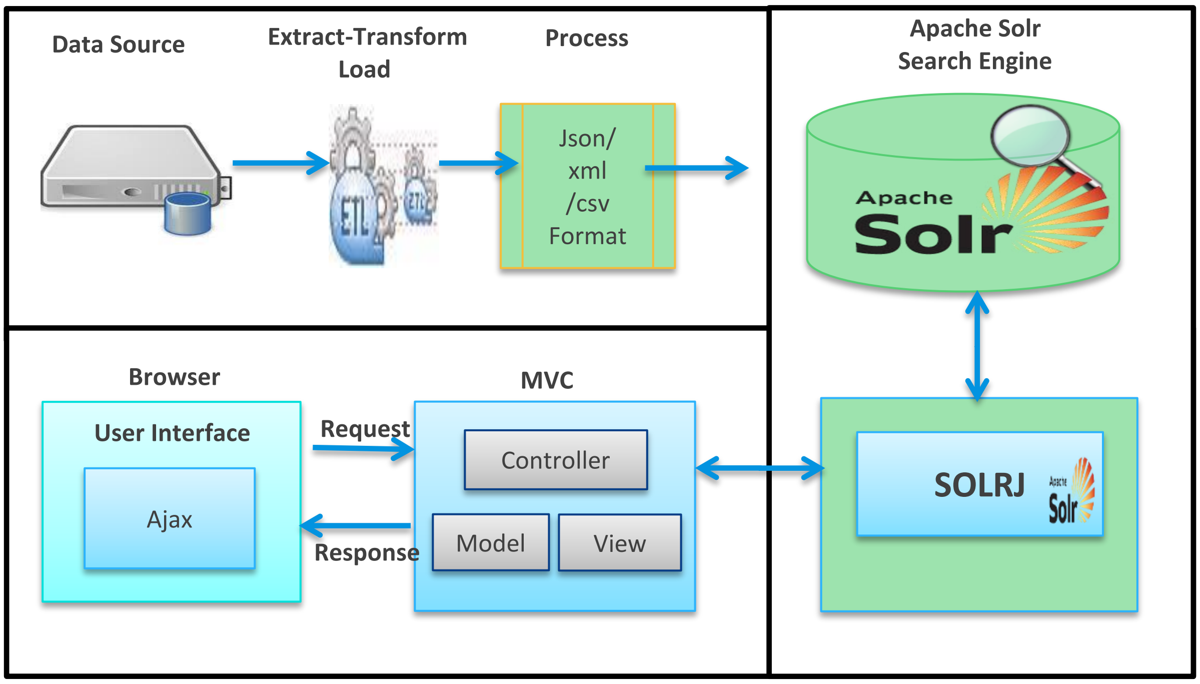
The solution overview of a standard ecommerce search application is as shown below. The most pre-dominant module of such an application is the search engine which indexes the product details and enable user to do a quick search and find products of their interest. Also a well-designed, simple yet intuitive graphic user interface is of most significance.

The diagram represented below (Figure 2) has three important modules. The first module is the Data Storage System which collates the data from file servers, shared folders, rich text documents, emails, data bases, and etc. The data undergoes an ETL Process i.e., data is extracted, transformed into a standard encoding format that Solr accepts, such as Json/XML/CSV and finally loaded into the Apache Solr. Solr search engine does the analyses on the input data and creates inverted indexes into its system. It also copies the indexes into different servers to handle auto failover mode and distributes indexes into multiple machine to balance the load. The third module that follows the model-view controller (MVC) pattern is primarily designed to handle user request from the browser and to send back the response in the acceptable format. There is a service layer also known as manager layer that will have the business logic to query the Solr search engine based on the user's search query. SolrJ is a java client to access solr. It offers a java interface to add, update, and query the solr index.

The MVC pattern is made up of three important components namely controller, model and view. The *controller* component is responsible for managing the entire request coming from the view or user interface. The view is responsible for displaying the data obtained from the model, and also to take the input from user and to pass it onto the controller. The model represents an entity object that provides the persistent storage of data obtained from data access layer.

Architecture: Java/J2EE Based Ecommerce Search Application

Figure 2



Apache Solr's features detailed

Auto Suggest:
A feature that assists the user by automatically predicting the remaining characters in a word or phrase based on what has been typed or input before. The terms/phrases are retrieved by using Solr's suggester component, facet component and terms component.

```
> facet=true&facet.prefix=keyWord,
> terms.fl=name&terms.lower=keyWord,
> suggest?q=keyWord and others
```

Results Highlighting:
When a user performs a search, words or phrases are highlighted in the results displayed making it easy for him to find the most relevant content. It is obtained by using *Solr's highlight component*.

```
> hl=true,hl.q=keyWord, hl.fl=field_names
> hl.simple.pre=</br>, hl.simple.post=</br> - These parameters
  can be used for appending html tags with the results list
```

Similar Keywords Suggestions:
Similar keywords are suggested based on the similarity of the keywords to the search terms a person enters while searching. Solr's *More Like This* component and synonym search filter can be used to enable this feature.

```
> mlt=true& mlt.fl=field Names&
> <filter class="solr.SynonymFilterFactory"/> - Filter should be
configured and auto-wired in the schema
```

Facet Navigation:
Faceted search lets the users to navigate through a collection of information by applying filters in whatever order they choose. Products are listed based on the categories they belong to and navigation is a way of refining a broad category of products in to what a user looking for. Solr has a highly featured Facet Component that provides prefix based facets, hierarchical facets, range facets and etc.

```
> Facet=true& facet.field=fieldNames -To obtain the standard facets
> facet.pivot for hirearchical faceting,
> facet.date for faceting on date field,
> facet.range on range queries and
> Many more.
```

© 2013 Happiest Minds Technologies Pvt. Ltd. All Rights Reserved

Filters:
Solr's fq parameter can be used to specify a filter query that filters the results from the super set of documents, without influencing score. It can be very useful for speeding up complex queries since the queries specified with fq are cached independently from the main query. Caching means the same filter is used again for a later query [from solr wiki].

```
> fq = fieldName:[condition],
> Example: fq = popularity:[10 TO *] AND status=Active
> q=*:*&fq={!cache=false}inStock:true - An example to cache filters
```

Sorting:
Sorting is organizing a collection of data elements based on the most-used orders namely numerical order and lexicographical order. Solr also has sorting interfaces which does sorting on numeric, range and alphabets.

```
> &sort=fieldName [sort type] ex: &sort=score desc, name asc
> Sorting can be applied at the time of query result caching and
document caching. This will give the better performance and the
cached results will be sorted for once and stored.
```

Promotional Products:
Products marked as Promotional can be retrieved based on keywords entered by the user. The technique is to have a field that indicates whether the product is promotional. Then apply the field level boosting to obtain these products on top of your result set.

```
> By default, a "TF-IDF" based Scoring Model is used for relevancy
scoring.
> To boost the score of new/promotional products, Apache Solr has
document level and field level boosting. Applying these
functionalities we can have solution approach to the above aspect.
> Example:q={!boost b=fieldName}text:keyword
> To give a negative (or very low) boost to documents that match a
query use the negate symbol and use caret symbol to give positive
boost.
> Ex: q = foo^100 bar^100 (*:* -xxx) ^999
```

Spell Check & Correction:
SpellcheckComponent is designed to provide inline spell checking. We can have file based spell checking or index based spell check. For spell-check, we need to add configurations in Solr Config.

```
> /spell? q=text:keyWord&spellcheck=true&spellcheck.collate=true&
> spellcheck.build=true&spellcheck.count=n
> Default spell-check applies default distance algorithms but in
case if you would need to have levenstian/jarowinkler distance
algorithms applied on spell-check, you can configure it separately.
```

happiest minds

© 2013 Happiest Minds Technologies Pvt. Ltd. All Rights Reserved

No Results Found:
This scenario can be handled in different ways by analyzing the user entered search criteria and keywords.

```
> If the user entered stream of characters which did not match the
existing set of keywords, we should have shingle filter applied on
keyword fields. Shingle breaks the words into streams of characters
and thus retrieves results based on characters chunks entered.
> If the user miss-spelled a term, we can query on spell-check
component to get suggested words and
> Also use synonym filter, incase needed, to check if the user
entered a synonymous term.
> Showing no results or random results is apprehensive, so have the
system show relevant/similar results.
```

Field Collapsing & Grouping:
Grouping is classifying the results set based on each category and returning top N documents per group.

```
> &q=keyWord&group=true&group.field=fieldNames
> &group.query=[]  &group.limit
> We can also have facets on group, set group limits, group sort and
```

Synonym Search Results:
Synonym is the keyword or a phrase that means exactly or nearly the same as keyword in the same language. System proposes synonyms for the searched words based on the synonym lists maintained in a custom file. This list can be modified as per requirements. Search logic is adopted to provide synonymous results for the search performed.

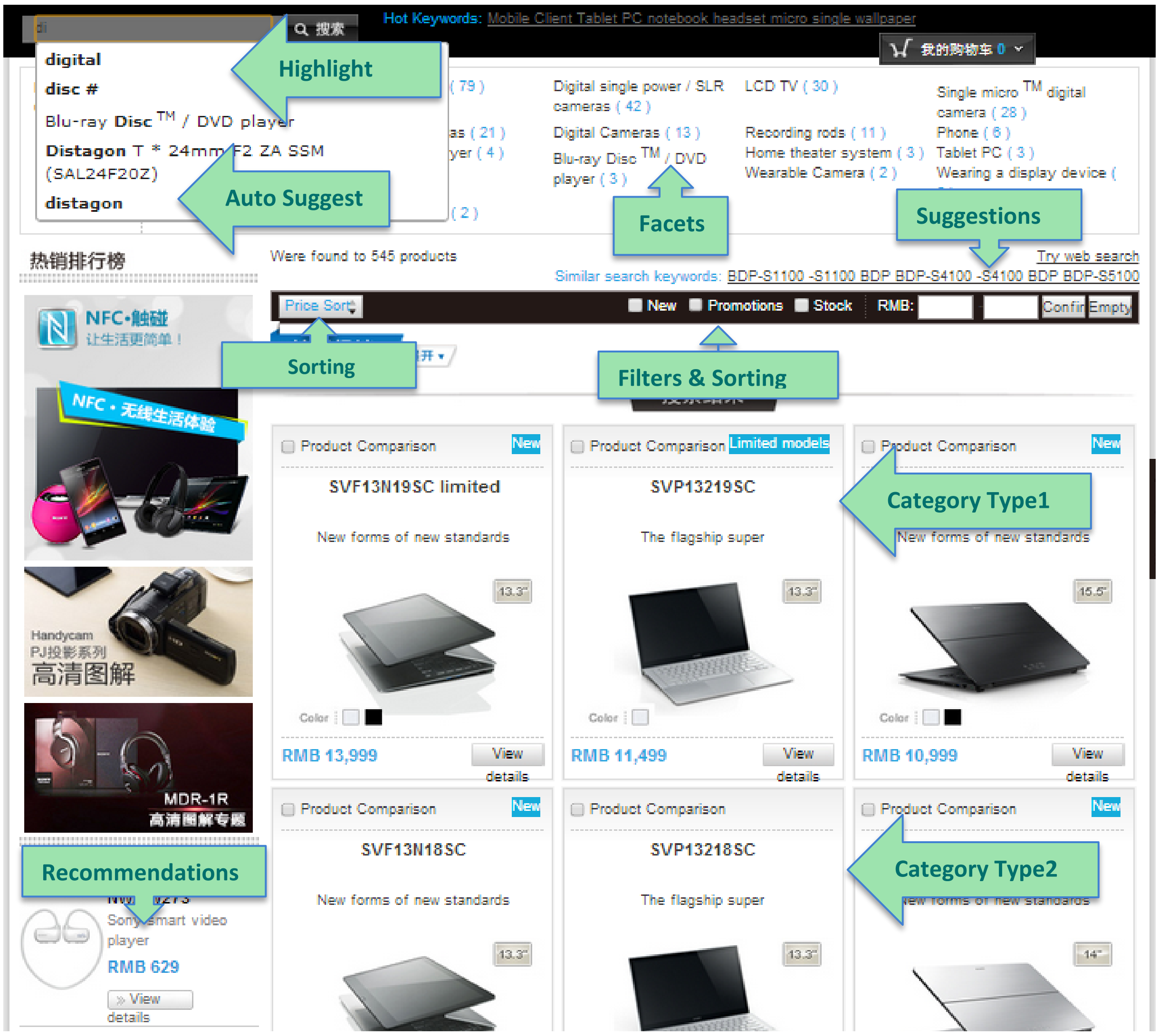
```
> Apache solr has a synonym based filter that can be wired to any data
type defined in the schema. With this filter, we also need to provide
a flat file with synonyms listed in it.
> Config glance : <filter class="solr.SynonymFilterFactory"
synonyms="syn.txt" ignoreCase="true" expand="false"/>. For a->b b->c
and c->a(cyclic match) set the expand attribute to true and for direct
match a->b,c , have the expand attribute set to false. [Refer wiki for
details]
```

Multi Lingual Support:
Multi Lingual Support: Data produced and consumed sometimes contain more than one language. Each langu age is unique, and presents different challenges to the search engine. Solr supports for multilingual search and cross-language information retrieval. [Refer to the presentation [12]]

```
There are APIs’ specific to languages that should be wired with data-
types. Find the list of languages supported in language analysis page
of Solr wiki [4].
```

© 2013 Happiest Minds Technologies Pvt. Ltd. All Rights Reserved

Sample Search Site [Figure 3 Image Ref Link: [10]]



Conclusion

Apache Solr is proven in terms of an agile search engine. It has all the core features requisite to build a great E-commerce search platform. Solr's scalability, its cloud features and flexibility enables the online retailers to easily adapt to meet the higher demands of traffic, and Lucene's search libraries ability can be leveraged to meet any retailer's requirements. A significant increase in the adoption of Solr and Lucene in search platforms is increasing interest in deploying the technology for ECommerce Search.

Also, Open source has many pragmatic effects, and Solr is no exception. It’s free/very low-cost and source plugins can be created as per business requirements. Apache Solr also has larger community of developers and committers. It is continually evolving in real time as developers add to it and modify it to make it a superior quality search platform.

© 2013 Happiest Minds Technologies Pvt. Ltd. All Rights Reserved

References

[1] ***TrechCrunch***, ***Forrester Forecast***, *2010*
http://techcrunch.com/2010/03/08/forrester-forecast-online-retail-sales-will-grow-to-250-billion-by-2014/

[2] ***depositphotos ,Icons List***
http://depositphotos.com/7917154/stock-photo-E-commerce-3D-icon--isolated-on-white.html

[3] ***Cominvent***, ***Norway***
http://www.cominvent.com/2011/04/04/solr-architecture-diagram/

[4] ***LuceneRevolution Conference, 2013***
http://www.lucenerevolution.org/2013/Lucene-Solr-Revolution-2013-Dublin-Presentations

[4] ***LuceneRevolution Conference, LinkedIn, 2013***
http://www.lucenerevolution.org/2013/How-Lucene-Powers-the-LinkedIn-Segmentation-and-Targeting-Platform

[5] ***SearchEngineJournal, Twitter***
http://www.searchenginejournal.com/twitters-search-function-now-powered-by-lucene/24780/

[6] ***LuceneRevolution Conference, Lucid Works***
http://www.lucenerevolution.org/2013/10-keys-to-Solrs-Future

[7] ***MikeMCCand***
http://people.apache.org/~mikemccand/lucenebench/indexing.html

[8] ***LuceneRevolution Conference***
http://www.lucenerevolution.org/2013/System-Teardown-Solr-as-a-Practical-Recommendation-Engine

[9] http://www.lucenerevolution.org/2013/Lucene-Solr-Revolution-2013-Dublin-Presentations

[10] www.sonystyle.com.cn/ps-web/

[12] ***Steve-Kearns, EuroCon 2011***
http://www.slideshare.net/lucenerevolution/steve-kearns-multilingualsearcheurocon2011

[13] ***LucidWorks***
http://docs.lucidworks.com/display/solr/Result+Grouping

[14] ***Happiest Minds Technologies, Rajani Maski, LuceneSolrConference, Dublin***
www.lucenerevolution.org/2013/A-Novel-methodology-for-handling-Document-Level Security-in-Search-Based-Applications

About Author:

At Happiest Minds, *Rajani Maski* is the Module Lead, specialized in Search Analytics and Solutions.
She is the Solution designer for enterprise Search softwares using open source technologies.
She also has strong knowledge base in data mining concepts and NoSQL platforms.
She was invited as a speaker at *LuceneSolr International Technical Conference* [14].
(mailto:rajani.maski@happiestsminds.com)

© 2013 Happiest Minds Technologies Pvt. Ltd. All Rights Reserved