APACHE SOLR

Open Source Search Platform

Background

- Six years of enterprise search consulting experience

- Search platforms are typically deployed within a company firewall
  - File Shares, Intranet Sites
  - SharePoint, Documentum
  - SAP, PLM, Legacy Applications

- Experience with several enterprise search commercial products

Agenda

- Introduce Apache Solr
  - Terminology, Concepts, History, Architecture and Features
- Index Population
  - Schema Design (schema.xml)
  - Feed Payloads
  - Apache Tika
- Index Query
  - Search Protocol
  - Response Payloads
  - Request Handlers (solrconfig.xml)
  - Search Components
- Search-Based Applications

Concepts & Terminology

Apache Lucene – is a full text search engine library written entirely in Java. Lucene is embedded with Solr.

Apache Solr – is an enterprise search platform written in Java. It exposes web services that can manage the lifecycle of documents in the index.

Document – is Lucene/Solr's primary unit of storage – representing a flat collection of fields (no nesting).

Field – definition consists of a name and configurable type (text, integer, double, date).

Core – separate index and configuration. A single server can support multiple cores and it is used for data partitioning. Supports multitenant applications.

Shard – Is a chunk of a larger index. They are created to scale an index horizontally across machines.

SolrCloud – refers to a set of features that enable your search index to be scaled across a cluster of nodes.

Concepts & Terminology

Synonyms – is a query expansion feature where (e.g. MB => megabyte)

Stop Words – are words that should be filtered from index storage and queries

Structured Content – refers to content that has been richly tagged with metadata.

Unstructured Content – MS Office, PDF documents, emails, instant messages, etc.

ACL – access control list used to capture document permissions

Early Binding – an authorization enforcement model where the document ACLs are stored in the index.

Late Binding – an authorization enforcement model where document authorization is not determined until query time.

ETL – extract (content source), transform (normalize the data), load (into index)

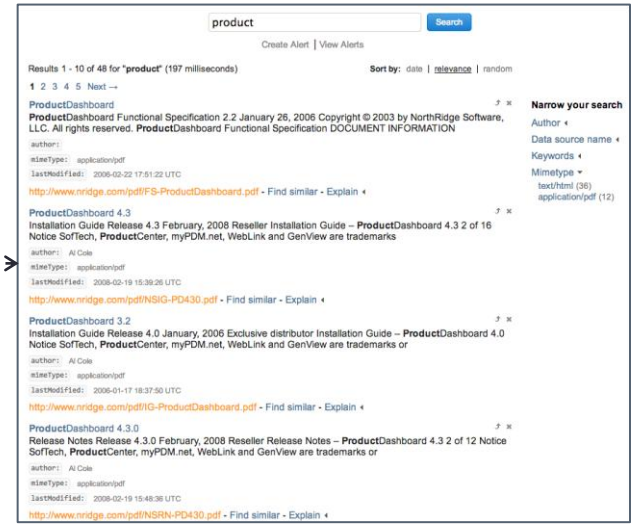
Search Based Application – built on top of search platforms and they are designed to deliver unified information access.

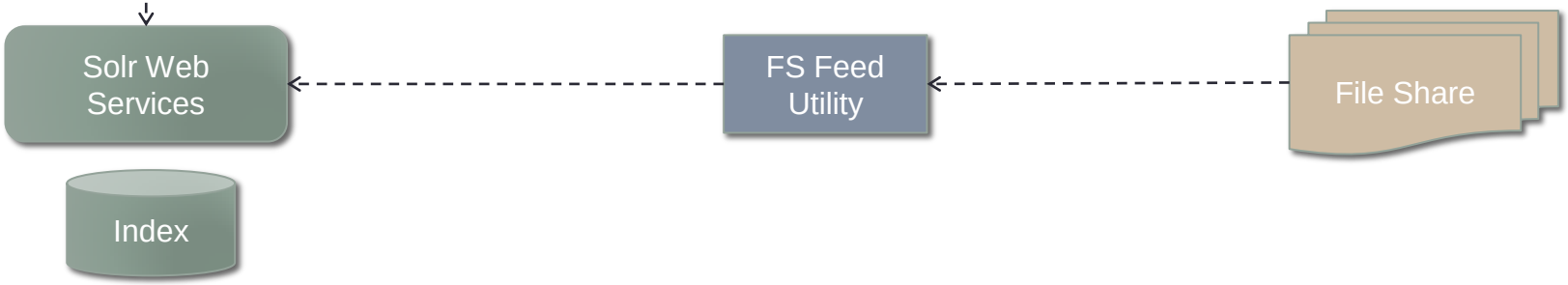
Lucene/Solr History

- Doug Cutting created Lucene in 1999
  - Recognized as a top level Apache Software Foundation project in 2005
- Yonik Seeley created Solr in 2004
  - Recognized as a top level Apache Software Foundation project in 2007
- Apache Lucene and Solr projects merge in 2010
  - Apache Lucene/Solr Release 1.4 in 2011
  - Apache Lucene/Solr Release 3.x in 2012
  - Apache Lucene/Solr Release 4.x in 2013

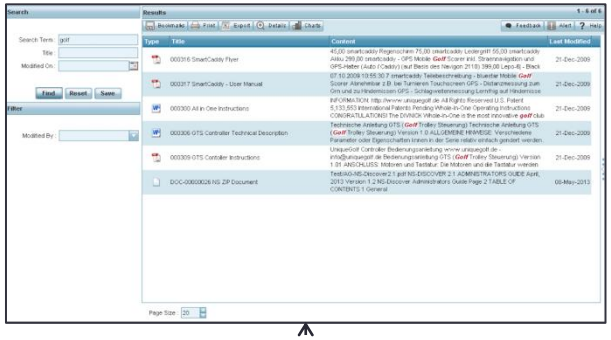
Sources: http://en.wikipedia.org/wiki/Lucene and http://en.wikipedia.org/wiki/Apache_Solr

Simple Search Architecture





Enterprise Search Architecture



Application Server

Solr Web Services

Index

FS Connector

File Share

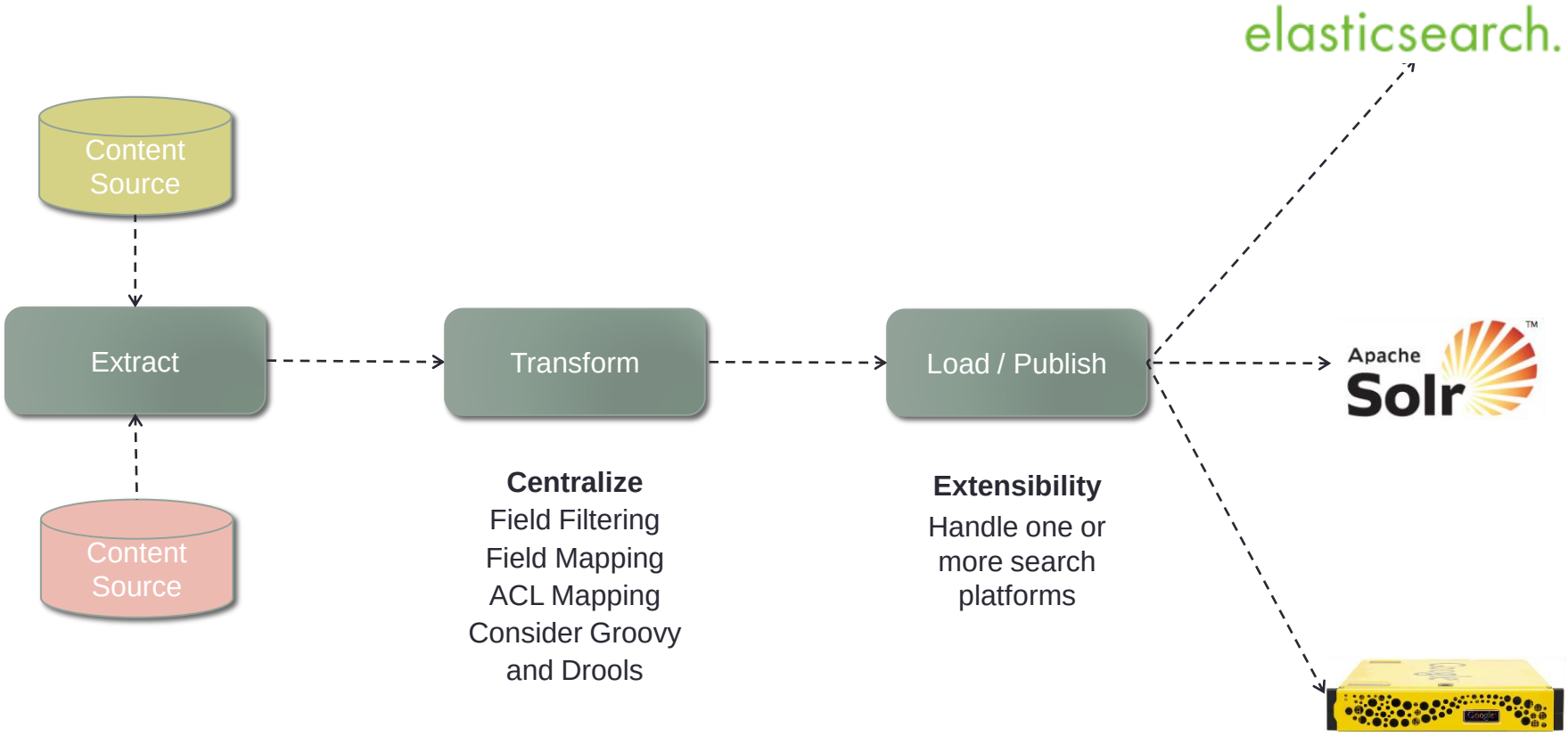
Application Connector

RDBMS

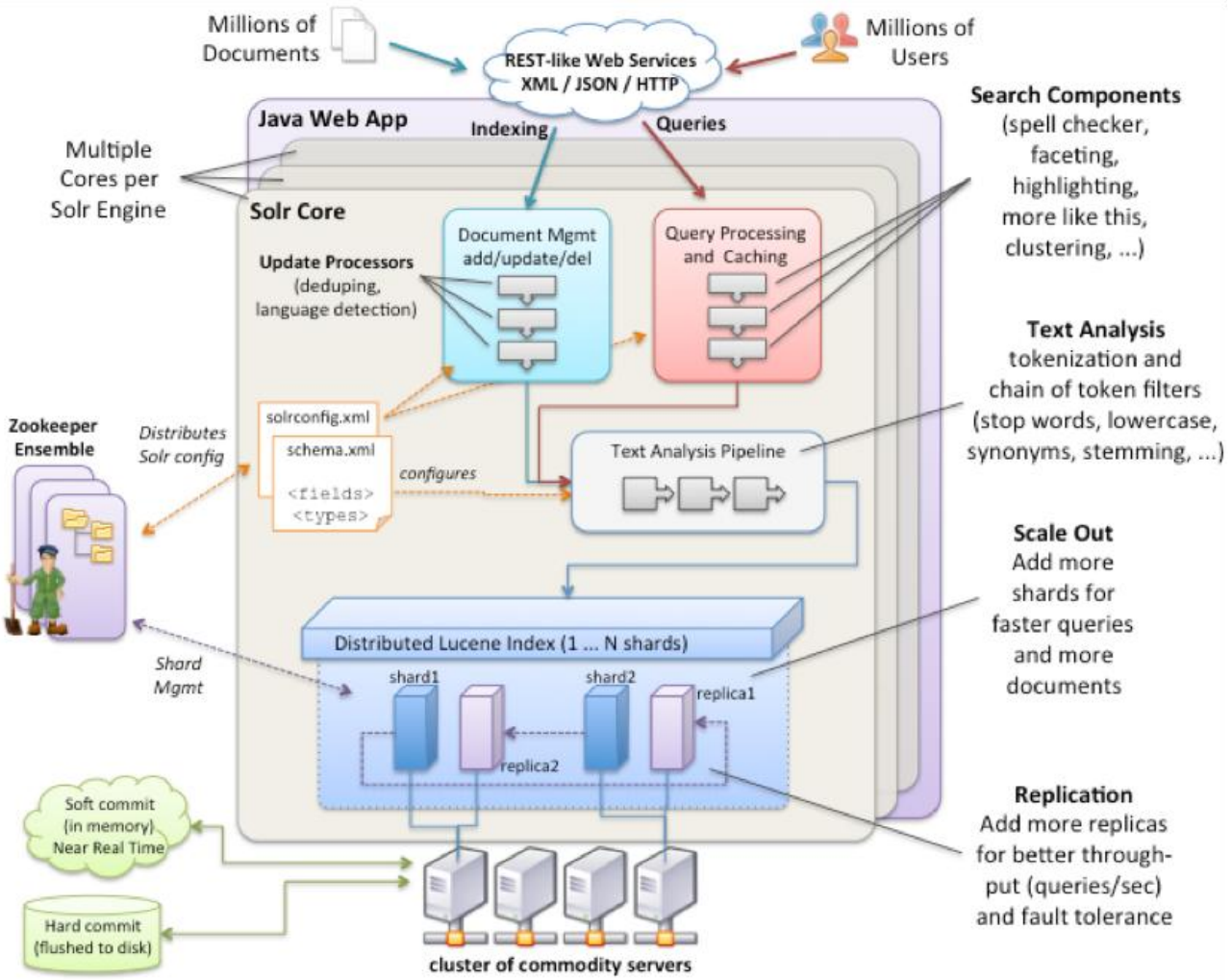
Web Site Connector

Web Site

ETL Process


elasticsearch.
Content Source
Extract
Content Source
Transform
Load / Publish
Apache Solr
Centralize
Field Filtering
Field Mapping
ACL Mapping
Consider Groovy and Drools
Extensibility
Handle one or more search platforms

Solr Architecture



Source: **Solr In Action**

Solr Features

Keyword Searching – queries of terms and boolean operators

Ranked Retrieval – sorted by relevancy score (descending order)

Snippet Highlighting – matching terms emphasized in results

Faceting – ability to apply filter queries based on matching fields

Paging Navigation – limits fetch sizes to improve performance

Result Sorting – sort the documents based on field values

Solr Features

Spelling Correction – suggest corrected spelling of query terms

Synonyms – expand queries based on configurable definition list

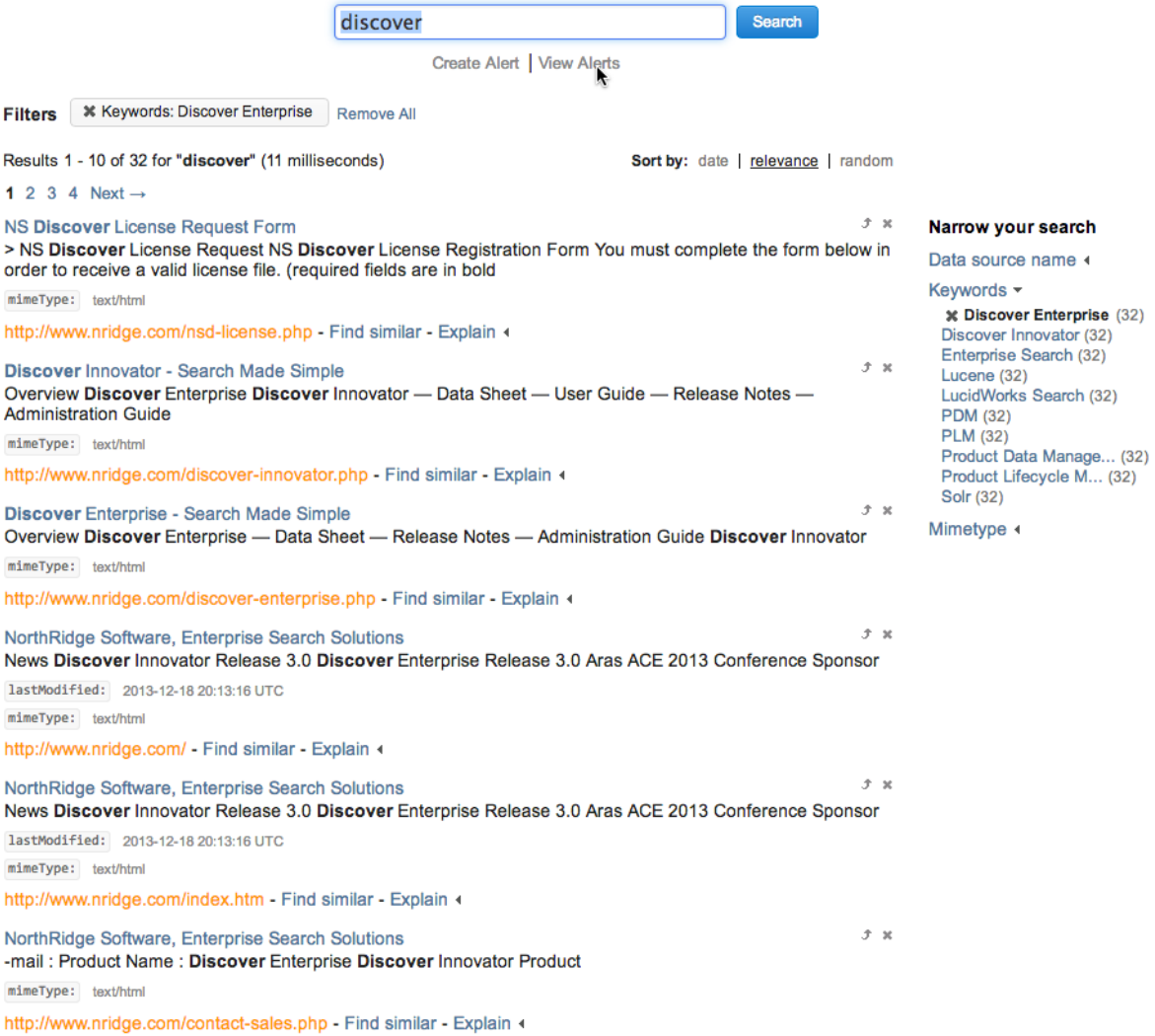
Auto-Suggestions – present list of possible query terms

More Like This – identifies other documents that are similar to one in a result set

Geo-Spatial Search – locate and sort documents by distance

Scalability – ability to break a large index into multiple shards and distribute indexing and query operations across a cluster of nodes

Solr Feature Example



Solr Installation

- Tutorial Available
  - https://lucene.apache.org/solr/4_6_1/tutorial.html
- Download
- Installation
- Index Population
  - Sample Documents
  - Feed Upload
  - Document Updates
  - Document Deletion
- Querying
  - Keywords
  - Facets

Solr Tutorial

Overview

This document covers the basics of running Solr using an example schema, and some sample data.

Requirements

To follow along with this tutorial, you will need...

1. Java 1.6 or greater. Some places you can get it are from Oracle, Open JDK, or IBM.
   - Running `java -version` at the command line should indicate a version number starting with 1.6.
   - Gnu's GCJ is not supported and does not work with Solr.
2. A Solr release.

Getting Started

Please run the browser showing this tutorial and the Solr server on the same machine so tutorial links will correctly point to your Solr server.

Begin by unzipping the Solr release and changing your working directory to be the "`example`" directory. (Note that the base directory name may vary with the version of Solr downloaded.) For example, with a shell in UNIX, Cygwin, or MacOS:

```
user:~solr$ ls
solr-nightly.zip
user:~solr$ unzip -q solr-nightly.zip
user:~solr$ cd solr-nightly/example/
```

Solr can run in any Java Servlet Container of your choice, but to simplify this tutorial, the example index includes a small installation of Jetty.

To launch Jetty with the Solr WAR, and the example configs, just run the `start.jar` ...

```
user:~/solr/example$ java -jar start.jar
2012-06-06 15:25:59.815:INFO:oejs.Server:jetty-8.1.2.v20120308
2012-06-06 15:25:59.834:INFO:oejdp.ScanningAppProvider:Deployment monitor .../solr/example/webapps at interval 0
2012-06-06 15:25:59.839:INFO:oejd.DeploymentManager:Deployable added: .../solr/example/webapps/solr.war
...
Jun 6, 2012 3:26:03 PM org.apache.solr.core.SolrCore registerSearcher
INFO: [collection1] Registered new searcher Searcher@7527e2ee main{StandardDirectoryReader(segments_1:1)}
```

This will start up the Jetty application server on port 8983, and use your terminal to display the logging information from Solr.

You can see that the Solr is running by loading http://localhost:8983/solr/ in your web browser. This is the main starting point for Administering Solr.

Indexing Data

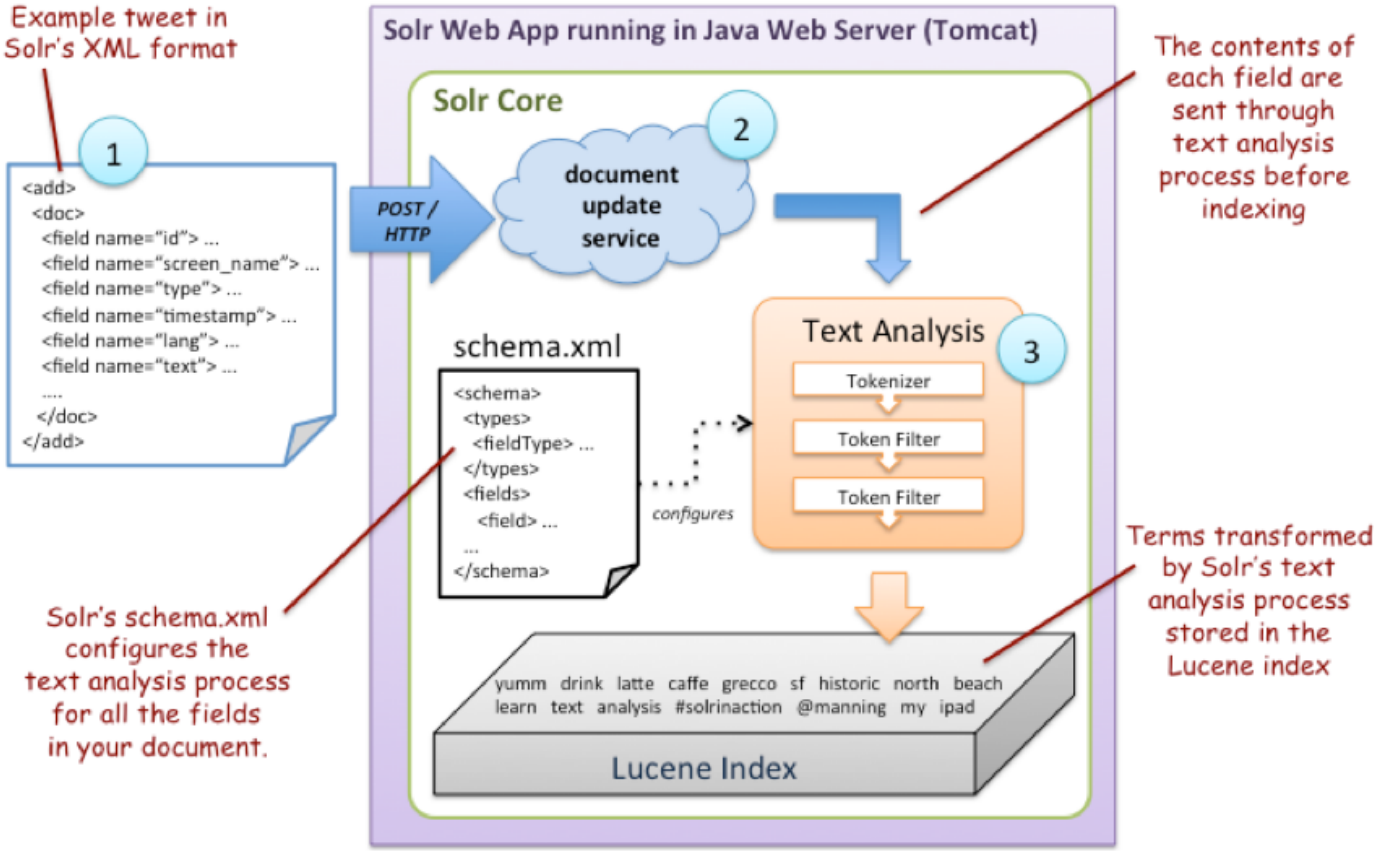
Your Solr server is up and running, but it doesn't contain any data. You can modify a Solr index by POSTing commands to Solr to add (or update) documents, delete documents, and commit pending adds and deletes. These commands can be in a variety of formats.

The `exampledocs` directory contains sample files showing of the types of commands Solr accepts, as well as a java utility for posting them from the command line (a `post.sh` shell script is also available, but for this tutorial we'll use the cross-platform Java client. Run `java -jar post.jar -h` so see it's various options).

Schema Document Design

- Information is captured in a document container.
- Each document consists of a list of fields.
- One field must uniquely identify each document in the index.
- Which fields will your users want to search on?
- What fields should be displayed in your search results?
- Structured versus unstructured content.
- Security model – public, ACLs, early versus late binding.

Indexing Process



Source: **Solr In Action**

Inverted Index

Original documents

Doc #	Content field
1	A Fun Guide to Cooking
2	Decorating Your Home
3	How to Raise a Child
4	Buying a New Car
5	Buying a New Home
6	The Beginner's Guide to Buying a House
7	Purchasing a Home
8	Becoming a New Home owner
9	How to Buy Your First House

Lucene's inverted index

Term	Doc #
a	1,3,4,5,6,7,8
becoming	8
beginner's	6
buy	9
buying	4,5,6
car	4
child	3
cooking	1
decorating	2
first	9
fun	1
…	…

(Continued)…	
…	…
guide	1,6
home	2,5,7,8
house	6,9
how	3,9
new	4,5,8
owner	8
purchasing	7
raise	3
the	6
to	1,6,9
your	2,9

Source: **Solr In Action**

Schema Configuration (schema.xml)

```
<schema name="example" version="1.5">
 <fields>
   <field name="id" type="string" indexed="true" stored="true" required="true" multiValued="false" />

   <field name="title" type="text_general" indexed="true" stored="true" multiValued="true"/>
   <field name="subject" type="text_general" indexed="true" stored="true"/>
   <field name="description" type="text_general" indexed="true" stored="true"/>
   <field name="comments" type="text_general" indexed="true" stored="true"/>
   <field name="author" type="text_general" indexed="true" stored="true"/>
   <field name="category" type="text_general" indexed="true" stored="true"/>
   <field name="last_modified" type="date" indexed="true" stored="true"/>
   <field name="links" type="string" indexed="true" stored="true" multiValued="true"/>
   <field name="content" type="text_general" indexed="false" stored="true" multiValued="true"/>
   <field name="text" type="text_general" indexed="true" stored="false" multiValued="true"/>

   <field name="weight" type="double" indexed="true" stored="true"/>
   <field name="price"  type="float" indexed="true" stored="true"/>
   <field name="popularity" type="int" indexed="true" stored="true" />
   <field name="inStock" type="boolean" indexed="true" stored="true" />
   <field name="store" type="location" indexed="true" stored="true"/>

   <dynamicField name="*_s"  type="string"  indexed="true"  stored="true" />
   <dynamicField name="*_dt"  type="date"    indexed="true"  stored="true"/>
 </fields>

 <uniqueKey>id</uniqueKey>

 <copyField source="title" dest="text"/>
 <copyField source="author" dest="text"/>
 <copyField source="description" dest="text"/>
 <copyField source="keywords" dest="text"/>
 <copyField source="content" dest="text"/>
```

Schema Configuration (schema.xml)

```
<types>
 <fieldType name="string" class="solr.StrField" sortMissingLast="true" />
 <fieldType name="boolean" class="solr.BoolField" sortMissingLast="true"/>
 <fieldType name="int" class="solr.TrieIntField" precisionStep="0" positionIncrementGap="0"/>
 <fieldType name="float" class="solr.TrieFloatField" precisionStep="0" positionIncrementGap="0"/>
 <fieldType name="long" class="solr.TrieLongField" precisionStep="0" positionIncrementGap="0"/>
 <fieldType name="double" class="solr.TrieDoubleField" precisionStep="0" positionIncrementGap="0"/>

 <fieldType name="text_general" class="solr.TextField" positionIncrementGap="100">
   <analyzer type="index">
    <tokenizer class="solr.StandardTokenizerFactory"/>
     <filter class="solr.StopFilterFactory" ignoreCase="true" words="stopwords.txt" />
     <filter class="solr.LowerCaseFilterFactory"/>
    </analyzer>
    <analyzer type="query">
     <tokenizer class="solr.StandardTokenizerFactory"/>
      <filter class="solr.StopFilterFactory" ignoreCase="true" words="stopwords.txt" />
      <filter class="solr.SynonymFilterFactory" synonyms="synonyms.txt" ignoreCase="true" expand="true"/>
      <filter class="solr.LowerCaseFilterFactory"/>
     </analyzer>
 </fieldType>
</types>
</schema>
```

Schema Design: Solr Unleashed Tutorial

Analyzers, Tokenizers and Filters: Solr Reference Documentation Solr Unleashed Tutorial

Document Text Extraction

NS-Discover Administrator's Guide

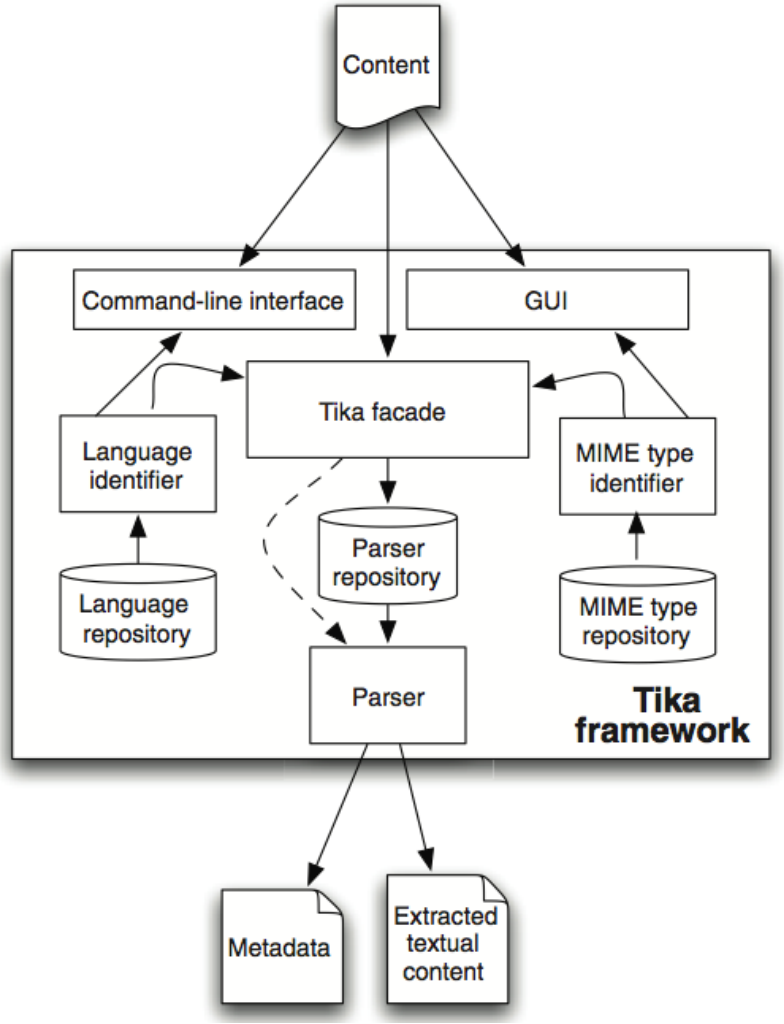
TABLE OF CONTENTS

GENERAL INFORMATION 1.1 OBJECTIVE The objective of this document is to provide the NS-Discover administrator with all the information required to install and maintain all NS-Discover products. This guide includes instruction for the installation and maintenance of: □ Discover Enterprise □ NS-Discover Index Server o Solr 4.1 Data Source □ Discover Innovator Connector □ Discover Innovator Client □ NS-Discover RDBMS Server 1.2 SUPPORTED FILE FORMATS The following table identifies all the file formats whose content will be included as searchable text from NS-Discover clients. FileType File Extension Notes MS Word .doc, .docx MS Excel .xls, .xlsx MS PowerPoint .ppt, .pptx MS Visio .vsd MS Project .ppt, .pptx Extracts the Title property only MS Publisher .pub HTML .html XML .xml Indexes arbitrary text Adobe Acrobat .pdf Plain Text .txt Rich Text Format .rtf Comma Separated Values .csv OpenXML .xml Microsoft's XML format ZIP Archive .zip Indexes archived file names only NS-Discover Administrator's Guide Page 5 2 INSTALLATION The Administrator is advised to review the entirety of this document in order to properly understand and be prepared for a successful installation. 2.1 PLATFORM SUPPORT The following table identifies the platforms upon which you can deploy and run the NS-Discover suite of products. Component OS Version Index Server Windows Server 2003, 2008 Windows XP, 7 Discover Innovator Connector Windows Server 2003, 2008 Windows XP, 7 Discover Innovator Client Internet Explorer 8, 9* * Please reference section 4.1.2 if you are running IE 9. 2.2 SERVER SIZING The Index Server is typically sized based on the number of documents it will process. However, there can be many other variables involved (e.g. queries per second, average document sizes, etc.), so the table below should be considered a starting point for hardware resource planning. Documents Disk Space (GB) RAM (GB) CPU Cores 10,000 .5 2 2 100,000 5 4 2 1,000,000 50 6 4 Although the number of users concurrently searching against the index will affect the performance, the estimates provided above should be sufficient to support hundreds of users. In addition, you should note that scaling vertically and horizontally is also possible with this architecture. Given this flexibility, we recommend keeping the initial configuration simple and deferring the use of a more sophisticated approach (e.g. master / slave machines, load balancers, etc.) until the production usage patterns justify that level of investment. NS-Discover Administrator's Guide Page 6 2.3 PREPARATION The following steps must be taken before beginning the installation process. 2.3.1 IDENTIFY HOST SERVERS NS-Discover is comprised of four server components, which can be installed on the same server or can be distributed amongst multiple servers. The following section provides guidance for installing the server components. 2.3.1.1 INDEX SERVER The NS-Discover Index Server is based on the Apache open source Lucene/Solr search platform. This component of the architecture is responsible for providing a repository for all the content in a centralized index, optimized for the search and retrieval of results. This platform is scalable to millions of documents and can be configured to meet specific customer requirements. The default installation of the Index Server includes the UI Server. If you are installing the NS-Discover Index Server on the same machine that hosts Aras Innovator, then there is no need to install the separate UI Server. 2.3.1.2 UI SERVER The NS-Discover UI Server hosts the client interface for Discover Innovator and must be installed on the Aras Innovator host machine. If the Index Server is installed on the same host as Aras Innovator, then the UI Server that is delivered with that installer will be used. If you install the Index Server on a host other than where Aras Innovator is installed, then a separate UI Server installation will be required on the Aras Innovator host machine. 2.3.1.3 DISCOVER INNOVATOR CONNECTOR The Discover Innovator Connector is responsible for extracting content from a populated Aras Innovator instance and feeding the data into the NS-Discover Index. The connector is also responsible for retrieving and opening files that have been requested by the user from the Discover Innovator Client. The Discover Innovator Connector must be installed on the same machine where the Innovator Vault Server resides. An Innovator account is required to install the Discover Innovator Connector. You must ensure that the Innovator account used for the installation has read access to all files stored within the Aras Innovator vault folders you wish to index. 2.3.1.4 RDBMS SERVICE The Discover RDBMS Service is required to support the following NS-Discover features. □ Discover User Portal o Saved Searches o Document Bookmarks □ Transaction monitoring and reporting NS-Discover Administrator's Guide Page 7 2.3.2 REGISTER AND DOWNLOAD SOFTWARE A license file for the Discover Innovator Connector can be obtained by completing the license request form on the NorthRidge Software web site. You can locate the license request form within the "Contact Us" menu on the main web page. Simply select the "License" menu entry and you will be directed to a request form like the one shown below. After you populate the form and submit it, you will receive a reply email message that contains: □ license file □ links to download the software □ links to the installation instructions NS-Discover Administrator's Guide Page 8 2.4 UPGRADE INSTALLATION PROCESS Follow these steps before running the NS-Discover installation program if you are upgrading from a previous version. This process will remove your existing index and repopulate the index with a full feed. Contact NorthRidge Software if you wish to preserve your existing index while upgrading to this release. 2.4.1 SHUTDOWN NSD SERVICES Open the Windows Services dialog from the server on which each of these services are currently running. They can be located on a single server or may distributed across servers. Stop each of the services before proceeding. □ NSD-AI □ NSD-Enterprise □ NSD-Index □ NSD-RDBMS The number of services running in your environment may vary depending on the products you currently have installed. 2.4.2 BACKUP THE CURRENT INSTALL FOLDERS Navigate to the current NS-Discover installation folder (NS_HOME). Copy the following folders to a separate backup folder. □ [NS_Home] \NSD-AI □ [NS_Home] \NSD-Index □ [NS_Home] \NSD-Enterprise □ [NS_Home] \NSD-RDBMS NS-Discover Administrator's Guide Page 9 2.4.3 UNINSTALL NS-DISCOVER

Apache Tika Framework

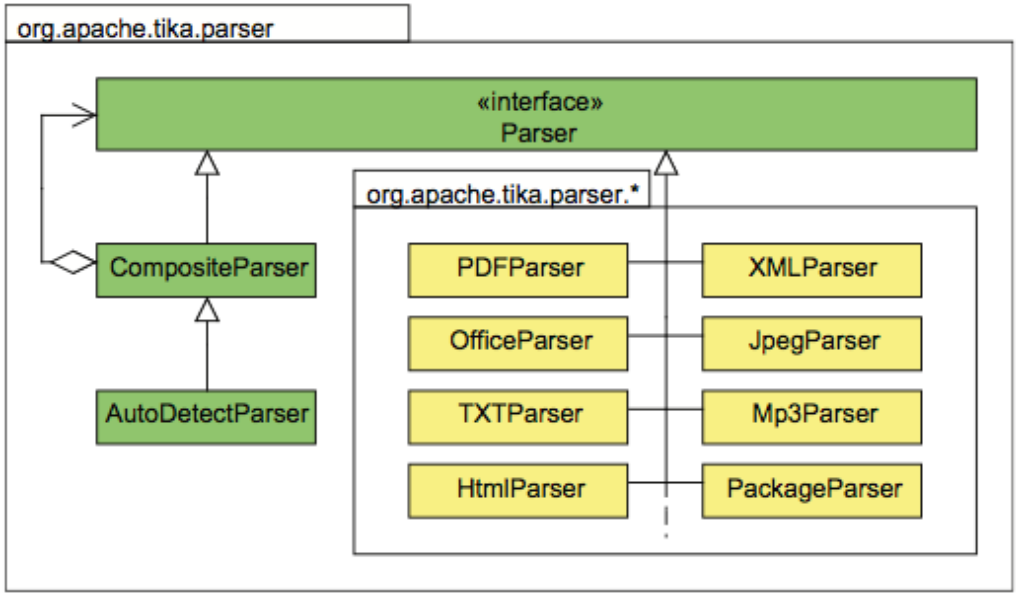
Supported Document Formats

- HyperText Markup Language
- XML and derived formats
- Microsoft Office document formats
- OpenDocument Format
- Portable Document Format
- Electronic Publication Format
- Rich Text Format
- Compression and packaging formats
- Text formats
- Audio formats
- Image formats
- Video formats
- Java class files and archives
- The mbox format



Source: **Tika In Action**

Apache Tika Framework




```
File document = new File("example.doc");
String content = new
Tika().parseToString(document);
System.out.print(content);
```

```
Parser tikaParser = new AutoDetectParser();
ParseContext parseContext = new ParseContext();
Parser recursiveMetadataParser = new RecursiveMetadataParser(new AutoDetectParser());
parseContext.set(Parser.class, recursiveMetadataParser);

WriteOutContentHandler writeOutContentHandler = new WriteOutContentHandler(aWriter, mMaxContentSize);
tikaParser.parse(inputStream, writeOutContentHandler, tikaMetaData, parseContext);
```

Source: **Tika In Action**

Solr Document

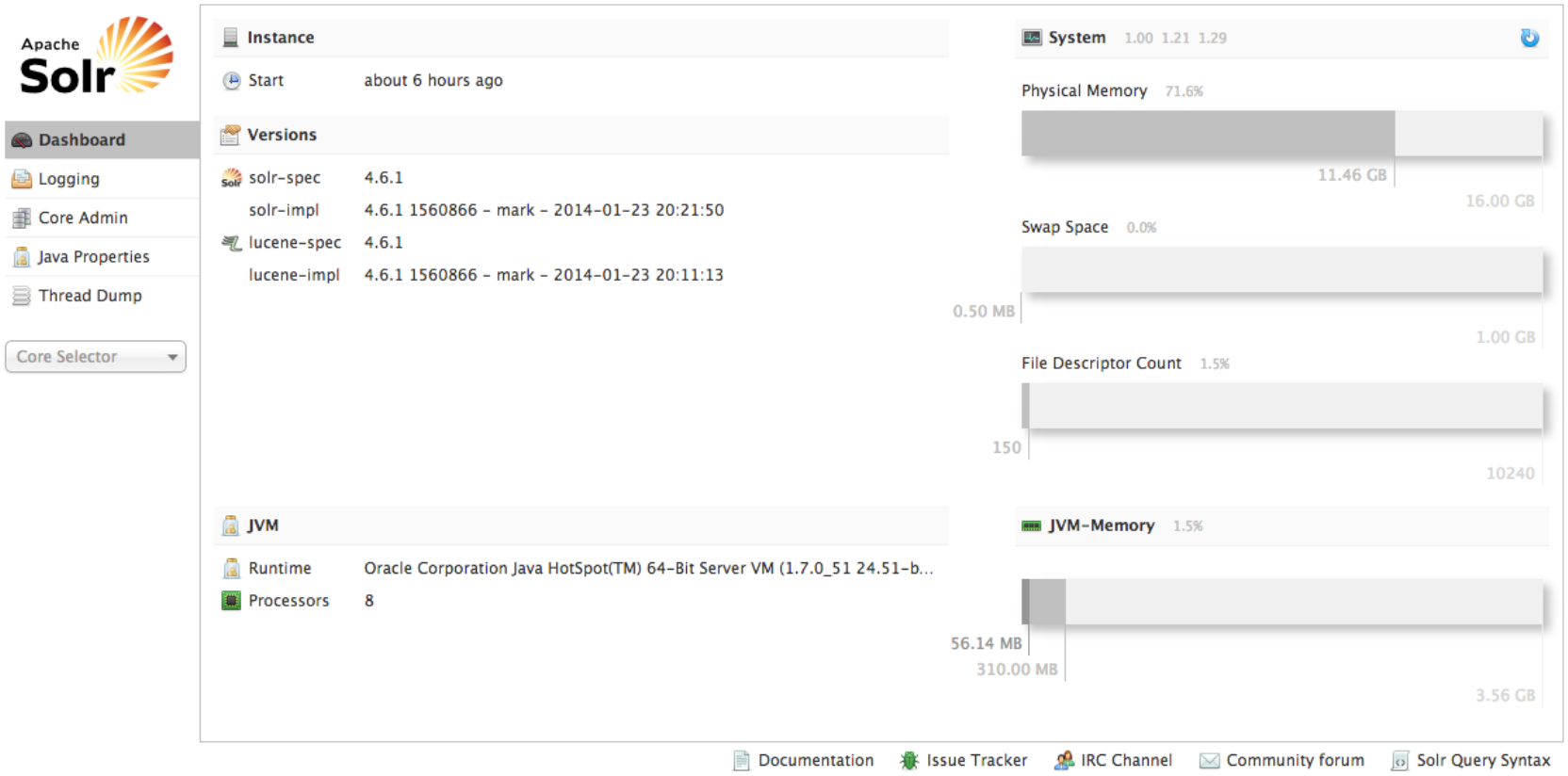
```
<add>
 <doc>
  <field name="id">SP2514N</field>
  <field name="name">Samsung SpinPoint P120 SP2514N - hard drive - 250 GB - ATA-133</field>
  <field name="manu">Samsung Electronics Co. Ltd.</field>
  <field name="cat">electronics</field>
  <field name="cat">hard drive</field>
  <field name="features">7200RPM, 8MB cache, IDE Ultra ATA-133</field>
  <field name="features">NoiseGuard, SilentSeek technology, Fluid Dynamic Bearing (FDB) motor</field>
  <field name="price">92</field>
  <field name="popularity">6</field>
  <field name="inStock">true</field>
  <field name="manufacturedate_dt">2006-02-13T15:26:37Z</field>
  <!-- Near Oklahoma city -->
  <field name="store">35.0752,-97.032</field>
 </doc>

 <doc>
  <field name="id">6H500F0</field>
  <field name="name">Maxtor DiamondMax 11 - hard drive - 500 GB - SATA-300</field>
  <field name="manu">Maxtor Corp.</field>
  <field name="cat">electronics</field>
  <field name="cat">hard drive</field>
  <field name="features">SATA 3.0Gb/s, NCQ</field>
  <field name="features">8.5ms seek</field>
  <field name="features">16MB cache</field>
  <field name="price">350</field>
  <field name="popularity">6</field>
  <field name="inStock">true</field>
  <!-- Buffalo store -->
  <field name="store">45.17614,-93.87341</field>
  <field name="manufacturedate_dt">2006-02-13T15:26:37Z</field>
 </doc>
</add>
```

SolrJ Library – Document Add

```
private void addSamplePart()
    throws IOException, SolrServerException
{
    HttpSolrServer solrServer = new HttpSolrServer("http://localhost:8983/solr/ce");
    solrServer.setParser(new XMLResponseParser());

    SolrInputDocument solrInputDocument = new SolrInputDocument();

    solrInputDocument.addField("id", "A1538104802");
    solrInputDocument.addField("datasheetTitle", "A) 4342, DATASHEET (2011/12/13)");
    solrInputDocument.addField("datasheetUrlLatest", "http://images.ihscontent.net/TXIIS133592-1.pdf");
    solrInputDocument.addField("objectId", "1538104802");
    solrInputDocument.addField("partCategory", "Operational Amplifier");
    solrInputDocument.addField("partDescription", "QUAD OP-AMP, 6000 uV OFFSET-MAX, 1 MHz BAND WIDTH");
    solrInputDocument.addField("partType", "Analog IC");
    solrInputDocument.addField("pubDate", "2011-12-13T00:00:00Z");
    solrInputDocument.addField("status", "ACTIVE");
    solrInputDocument.addField("manufacturerName", "EXAS INSTRUMENTS INC");
    solrInputDocument.addField("packageShape", "RECTANGULAR");
    solrInputDocument.addField("packageStyle", "SMALL OUTLINE");
    solrInputDocument.addField("technology", "CMOS");
    solrInputDocument.addField("terminalFinish", "NICKEL PALLADIUM GOLD");

    UpdateResponse updateResponse = solrServer.add(solrInputDocument);
    if (updateResponse.getStatus() == 0)
        solrServer.commit();
}
```

Tutorial: https://wiki.apache.org/solr/Solrj

Solr Dashboard



http://localhost:8983/solr/admin

Query Parameters

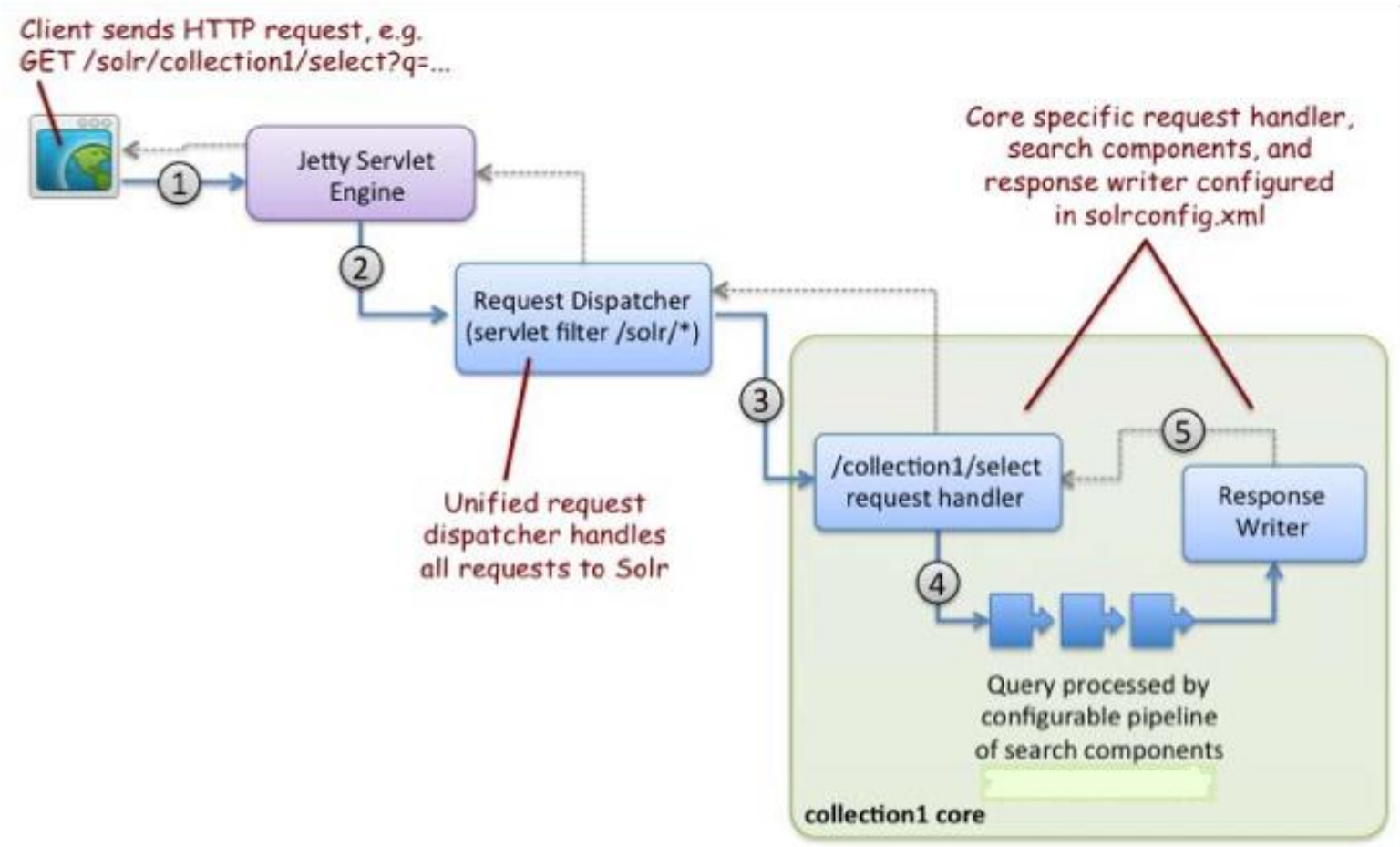
Parameter	Description
q	Main query parameter; documents are scored by their similarity to terms in this parameter.
fq	Filter query; restricts the result set to documents matching this filter but doesn't affect scoring.
start	Specifies the starting offset for a page for results; uses 0-based indexing. Start should be incremented by the page size to advance to the next page.
rows	Page size; restricts the number of results returned per page.
sort	Specifies the sort field and sort order; supports ascending (asc) and descending (des).
fl	List of fields to return for each document in the result set.
wt	Response-writer type; governs the format of the response.

Query Parsers: https://cwiki.apache.org/confluence/display/solr/The+Extended+DisMax+Query+Parser

Query Syntax Examples

Equal	title:discover title:”discover enterprise”
Not Equal	-title:discover
In Set	id:(100 OR 200 OR 300)
Not In Set	-id:(100 OR 200 OR 300)
String Data Type	
Starts With	title:discover*
Contains	title:*discover*
Ends With	title:*discover
Numeric Data Type	
Greater Than	price:[100 TO *]
Less Than	price:[* TO 100]
Between	price:[100 TO 500]
Not Between	-price:[100 TO 500]

Index Query



Source: **Solr In Action**

Request Configuration (solrconfig.xml)

```
<requestHandler name="/browse" class="solr.SearchHandler">
   <lst name="defaults">
     <str name="echoParams">explicit</str>

     <!-- VelocityResponseWriter settings -->
     <str name="wt">velocity</str>
     <str name="v.template">browse</str>
     <str name="v.layout">layout</str>
     <str name="title">Solritas</str>

     <!-- Query settings -->
     <str name="defType">edismax</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
        title^10.0 description^5.0 keywords^5.0 author^2.0 resourcename^1.0
     </str>
     <str name="df">text</str>
     <str name="mm">100%</str>
     <str name="q.alt">*:*</str>
     <str name="rows">10</str>
     <str name="fl">*,score</str>

     <str name="mlt.qf">
       text^0.5 features^1.0 name^1.2 sku^1.5 id^10.0 manu^1.1 cat^1.4
       title^10.0 description^5.0 keywords^5.0 author^2.0 resourcename^1.0
     </str>
     <str name="mlt.fl">text,features,name,sku,id,manu,cat,title,description,keywords,author,resourcename</str>
     <int name="mlt.count">3</int>
```

Request Handlers: https://wiki.apache.org/solr/SolrRequestHandler

Request Configuration

```
    <str name="mlt.qf">
      text^0.5 features^1.0 name^1.2 sku^1.5 id^10.0 manu^1.1 cat^1.4
      title^10.0 description^5.0 keywords^5.0 author^2.0 resourcename^1.0
    </str>
    <str name="mlt.fl">text,features,name,sku,id,manu,cat,title,description,keywords,author,resourcename</str>
    <int name="mlt.count">3</int>

    <!-- Faceting defaults -->
    <str name="facet">on</str>
    <str name="facet.field">cat</str>
    <str name="facet.field">manu_exact</str>
    <str name="facet.field">content_type</str>
    <str name="facet.mincount">1</str>

    <!-- Highlighting defaults -->
    <str name="hl">on</str>
    <str name="hl.fl">content features title name</str>
    <str name="hl.encoder">html</str>
    <str name="hl.simple.pre">&lt;b&gt;</str>
    <str name="hl.simple.post">&lt;/b&gt;</str>

    <!-- Spell checking defaults -->
    <str name="spellcheck">on</str>
    <str name="spellcheck.extendedResults">false</str>
    <str name="spellcheck.count">5</str>
  </lst>

  <!-- append spellchecking to our list of components -->
  <arr name="last-components">
    <str>spellcheck</str>
  </arr>
</requestHandler>
```

Request Handlers: https://cwiki.apache.org/confluence/display/solr/Searching

SolrJ Library – Document Query

```
private void querySamplePart()
    throws IOException, SolrServerException
{
    HttpSolrServer solrServer = new HttpSolrServer("http://localhost:8983/solr/ce");
    solrServer.setParser(new XMLResponseParser());

    SolrQuery solrQuery = new SolrQuery();
    solrQuery.setRequestHandler("getPart");

    solrQuery.setQuery("QUAD 6000");
    solrQuery.addFilterQuery("technology:CMOS", "status:ACTIVE");
    solrQuery.setFields("id", "datasheetTitle", "partCategory", "partDescription", "manufacturerName");
    solrQuery.setStart(0);
    solrQuery.set("defType", "edismax");

    QueryResponse queryResponse = solrServer.query(solrQuery);
    SolrDocumentList docResults = queryResponse.getResults();
    for (int i = 0; i < docResults.size(); ++i)
        System.out.println(docResults.get(i));
}
```

Tutorial: https://wiki.apache.org/solr/Solrj

Solritas

http://localhost:8983/solr/collection1/browse

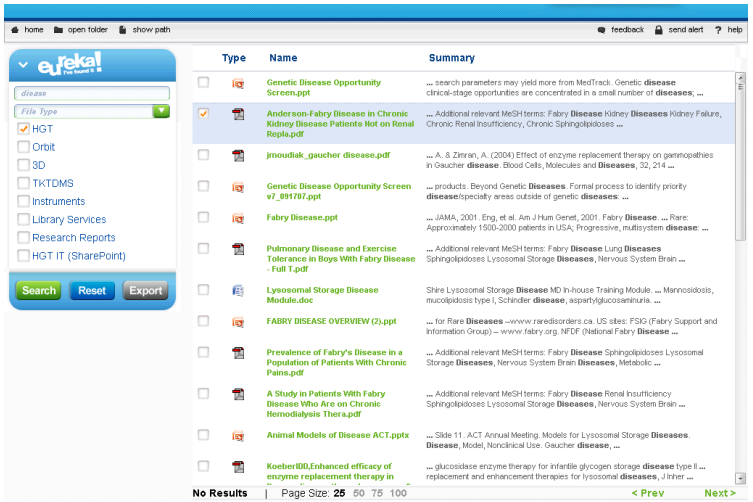
Search-Based Applications

Intranet Portal

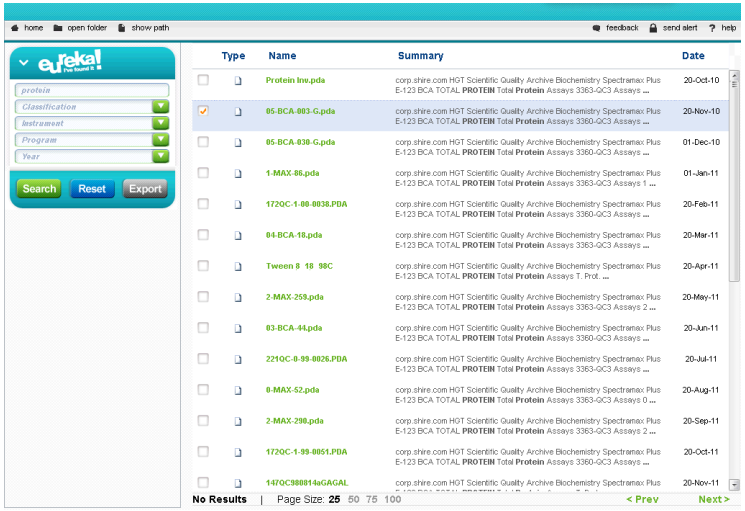
- Easy access to search
- News and event notification
- Single sign-on authentication
- Application launching

Federated Client

- Search across all content
- Authorized access only
- Simplified presentation
- Document viewing



Search Based Applications

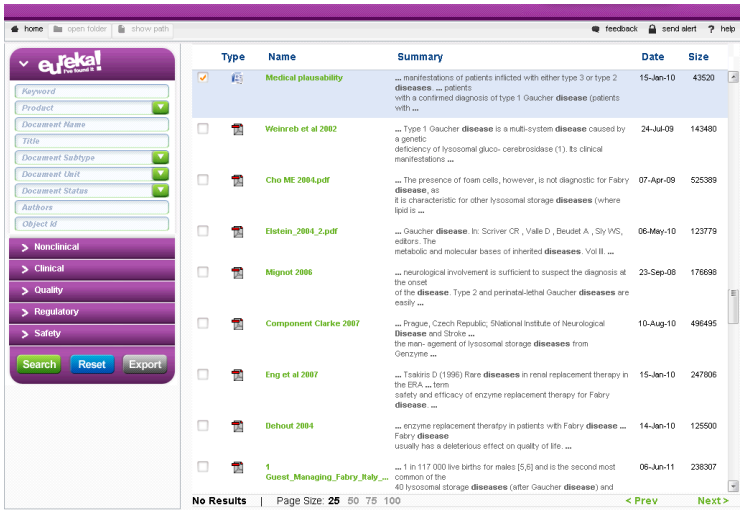


Instrument Datasets

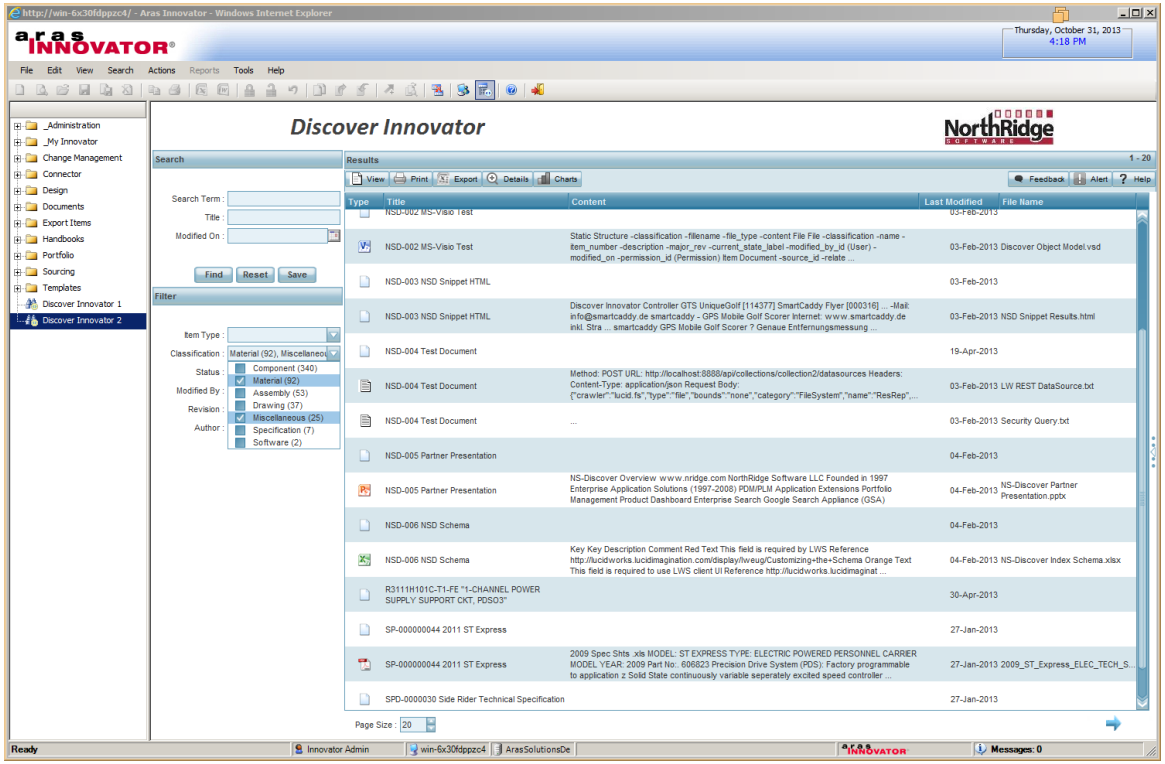
- Optimized for scientists
- Data dependent menus
- Specialized grid filters

Regulatory Documents

- Designed for researchers
- Rich meta-data access
- Spreadsheet exports
- View document accelerator



Search Based Applications



Embedded in PLM Application

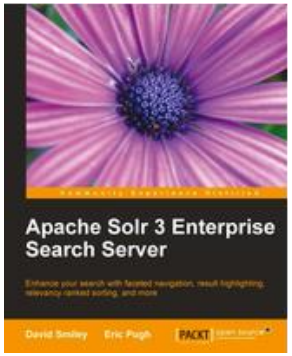
- Substantially better search experience than an RDBMS could provide
- Late-binding security model
- Document actions exposed on toolbar

Solr Resources

http://wiki.apache.org/solr/FrontPage

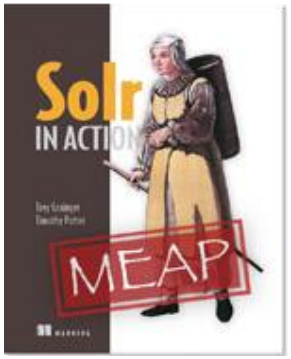
http://wiki.apache.org/solr/SolrResources

https://cwiki.apache.org/confluence/display/solr/



Apache Solr 3 Enterprise Search Server

- David Smiley and Eric Pugh
- Packt Publishing



Solr In Action

- Trey Grainger and Timothy Potter
- Manning Publications

Thank You!

Al Cole

acole@nridge.com

www.linkedin.com/in/coleal