

Progetto di
Ingegneria del Software A
e
Reti di Calcolatori A

Anno Accademico 2009/2010



Docente: Prof. Agostino Poggi

Studente: Davide Valeriani

Matricola 190883

Indice

Solr

Descrizione _____	3
Caratteristiche principali _____	4
Requisiti _____	6
Installazione _____	6
Utilizzo _____	8

RMS – Rapid Music Search

Descrizione _____	11
Requisiti _____	11
Installazione _____	12
Utilizzo _____	14
Conclusioni _____	21
Bibliografia _____	21

Che cos'è Solr

Solr (si pronuncia "Solar") è un "Enterprise Search Server", ovvero una piattaforma di ricerca aziendale open source basata sulla libreria Java per motori di ricerca Apache Lucene. Viene utilizzato per le ricerche nella maggior parte dei grandi siti web mondiali, come <http://www.cnet.com> o <http://sourceforge.net> e addirittura <http://www.whitehouse.gov>, in quanto garantisce una velocità di ricerca vicina alla ricerca nella rete locale Intranet.

Solr opera tra il Database e il Web, permette cioè ai siti web di interrogare il database in modo rapido ed efficiente (fig.1).



fig.1: Esempio di utilizzo di Solr

Solr è scritto in Java e lo stesso Java può essere utilizzato per modificare o estendere la piattaforma software. Viene eseguito come un server di ricerca indipendente all'interno di un contenitore servlet (applicazione eseguita su un server web) come Tomcat. Il potente sistema di configurazione esterno permette di adattare Solr alla maggior parte delle applicazioni senza la scrittura di codice Java.

Analizziamo ora più nel dettaglio il cuore di Solr, descrivendo Apache Lucene.

Lucene

Lucene è una libreria per motori di ricerca di testi open source e altamente performante. È stata sviluppata da Doug Cutting nel

2000 e si è evoluta da allora grazie al contributo di una forte comunità online. Essendo solamente una libreria di codice, Lucene non è un server. Pertanto, per essere utilizzata qualcuno deve scrivere il codice per memorizzare e interrogare un indice memorizzato su un disco.

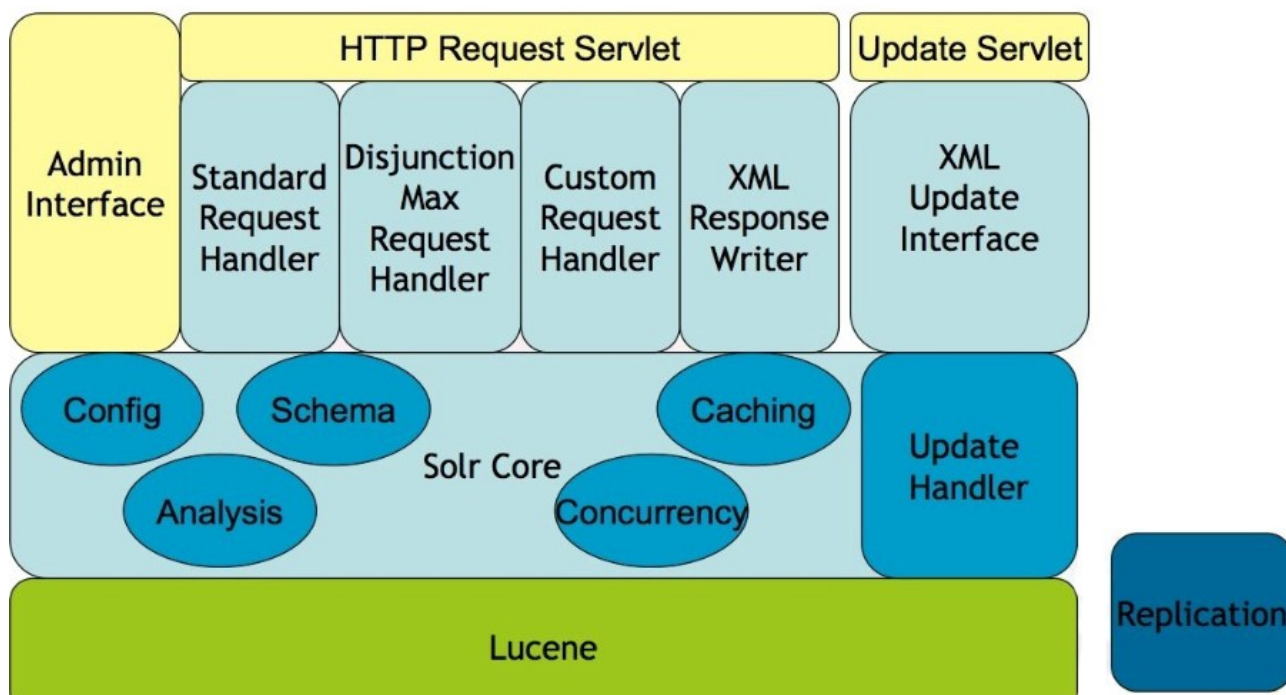


fig. 2: La struttura di Solr

Così, nel 2006 nasce Solr, che si può vedere come la trasformazione in server della libreria Lucene. Tuttavia, non è sottile il confine tra Solr e Lucene: infatti, molte caratteristiche e funzioni sono differenti, come la sfaccettatura.

Caratteristiche principali

Oltre alla caratteristica principale, ovvero una potente ricerca full-text, Solr offre numerose funzioni:

- elaborazione delle richieste HTTP per l'indicizzazione e la ricerca di documenti;

- numerose cache per risposte veloci alle interrogazioni;
- un'interfaccia amministrativa web-based che include:
 - statistiche delle prestazioni in fase di esecuzione;
 - un modulo di ricerca nell'indice;
 - un visualizzatore dello schema con istogrammi dei termini più comuni con alcune statistiche;
- file di configurazione per lo schema (descrizione dei dati) e il server stesso in XML:
 - Solr estende la libreria di analisi del testo di Lucene rendendola configurabile in XML;
 - introduce la nozione di "tipo di campo", utili per le date e speciali algoritmi di ordinamento;
- sfaccettatura dei risultati della ricerca (utile nei siti di e-commerce);
- permette di evidenziare i risultati;
- un plugin di controllo dello spelling, usato per suggerire query alternative (come il "forse cercavi" di Google);
- un plugin per elencare i documenti simili ad un documento scelto;
- un modello di server Solr distribuito con supporto di script per consentire uno sviluppo di larga scala;
- clustering dinamico, ovvero selezione e raggruppamento di elementi omogenei in un insieme di dati;
- integrazione nelle basi di dati;
- gestione di documenti di testo ricco, come i documenti scritti con un Word Processor (Write, Word, ...) o i PDF (Portable Document Format);

- altamente scalabile, ovvero può facilmente incrementare le proprie prestazioni se si aumentano le risorse a disposizione.

Requisiti

Nonostante sia scritto in Java, non è necessaria una conoscenza del linguaggio per utilizzare Solr, mentre per estenderlo è ovviamente fondamentale la sua padronanza.

Per utilizzare Solr è necessario avere installati i seguenti software:

- Java Development Kit (JDK) v1.5 o successive: fondamentale per l'esecuzione di Solr è la JRE, ma la JDK è necessaria per la compilazione da sorgente e l'estensione;
- Apache Ant, necessario per le modifiche di Solr, opzionale per l'utilizzo "base";
- un web server Java: Solr ne include già uno predefinito, Jetty, ma nel mondo reale spesso occorre utilizzarne di diversi.

Installazione

Dal sito <http://lucene.apache.org/solr/> è possibile scaricare l'archivio di file precompilati di Solr. Una volta scaricati, estraiamoli dall'archivio: verrà creata la cartella `apache-solr-1.4.0`.

Per vedere Solr in azione, dobbiamo avviare il web server Java. In questo caso, useremo quello di default già integrato nell'archivio Solr: Jetty. Per avviarlo, apriamo un terminale, entriamo nella cartella in cui abbiamo estratto Solr e digitiamo:

```
cd example
java -jar start.jar
```

A questo punto, verranno visualizzate una serie di stringhe di

output, fino all'ultima che avvisa che la connessione alla socket è stata avviata e, pertanto, Jetty è in esecuzione (fig.3).

```
29-dic-2009 13:40:46 org.apache.solr.core.SolrResourceLoader locateSolrHome
INFO: JNDI not configured for solr (NoInitialContextEx)
29-dic-2009 13:40:46 org.apache.solr.core.SolrResourceLoader locateSolrHome
INFO: solr home defaulted to 'solr/' (could not find system property or JNDI)
29-dic-2009 13:40:46 org.apache.solr.servlet.SolrServlet init
INFO: SolrServlet.init() done
29-dic-2009 13:40:46 org.apache.solr.core.SolrCore execute
INFO: [] webapp=null path=null params={start=0&event=firstSearcher&q=solr+rocks&rows=10} hits=0 status=0 QTime=49
29-dic-2009 13:40:46 org.apache.solr.core.SolrResourceLoader locateSolrHome
INFO: JNDI not configured for solr (NoInitialContextEx)
29-dic-2009 13:40:46 org.apache.solr.core.SolrResourceLoader locateSolrHome
INFO: solr home defaulted to 'solr/' (could not find system property or JNDI)
29-dic-2009 13:40:46 org.apache.solr.servlet.SolrUpdateServlet init
INFO: SolrUpdateServlet.init() done
29-dic-2009 13:40:46 org.apache.solr.core.SolrCore execute
INFO: [] webapp=null path=null params={event=firstSearcher&q=static+firstSearcher+warming+query+from+solrconfig.xml}
29-dic-2009 13:40:46 org.apache.solr.core.QuerySenderListener newSearcher
INFO: QuerySenderListener done.
29-dic-2009 13:40:46 org.apache.solr.handler.component.SpellCheckComponent$SpellCheckerListener newSearcher
INFO: Loading spell index for spellchecker: default
29-dic-2009 13:40:46 org.apache.solr.core.SolrCore registerSearcher
INFO: [] Registered new searcher Searcher0b40ec4 main
2009-12-29 13:40:46.527:INFO: Started SocketConnector @ 0.0.0.0:8983
```

fig. 3: Output finale del comando di avvio di Jetty

L'indirizzo 0.0.0.0 indica che la socket è in ascolto da ogni host, mentre 8983 indica la porta. Per terminare l'esecuzione di Jetty, è sufficiente premere Ctrl+C sulla tastiera nella finestra del terminale. Ora apriamo il browser e a visitare l'indirizzo locale:

http://localhost:8983/solr/admin/

Appare la pagina di amministrazione di Solr (fig.4). Vediamo ora come utilizzarlo.

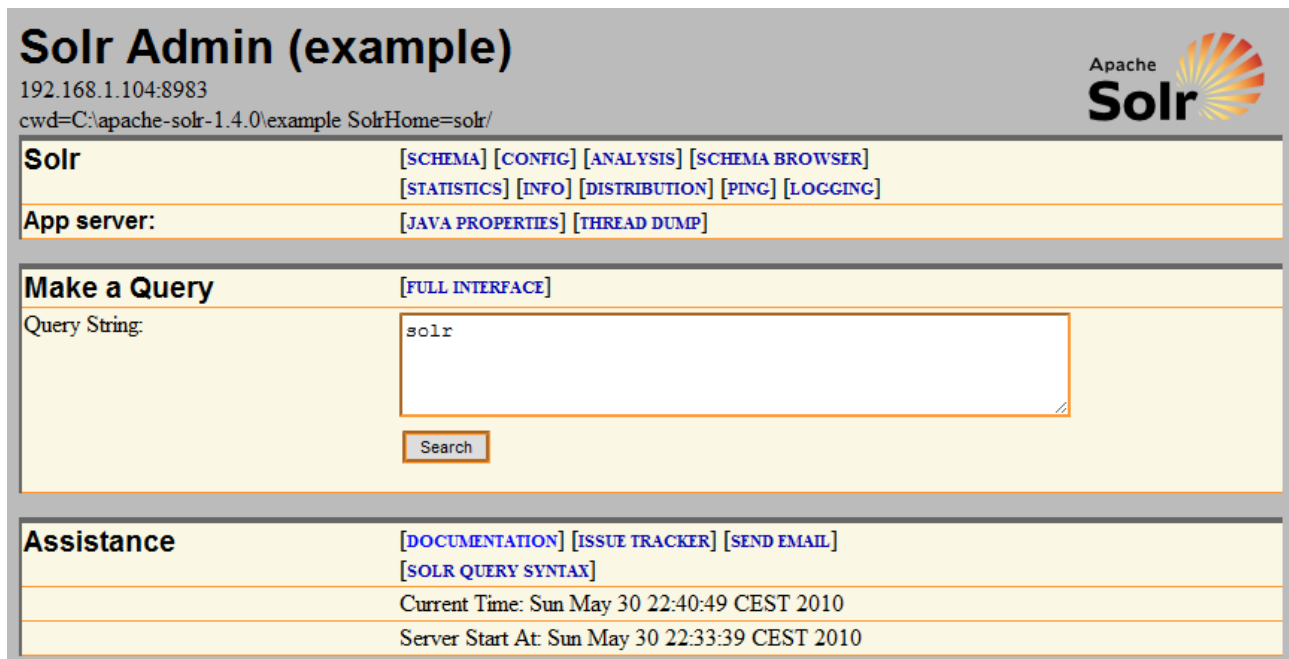


fig. 4 - Interfaccia amministrativa di Solr

Utilizzo

Per prima cosa, dobbiamo inserire dei dati all'interno del nostro database. Ci sono vari modi per inserire dati in Solr:

- Importarli da una database
- Caricare un file .CSV, esportato, ad esempio, da Excel o MySQL
- Inserirli da terminale attraverso file .xml

Noi utilizzeremo quest'ultimo metodo, in quanto quello più semplice che sfrutta, appunto, la semplicità del linguaggio XML.

Apriamo un editor di testo e scriviamo il seguente codice di esempio (i dati di un Apple iPhone):

```
<add><doc>
  <field name="id">123456</field>
  <field name="name">Apple iPhone 32 GB</field>
  <field name="manu">Apple Computer Inc.</field>
  <field name="cat">elettronica</field>
  <field name="cat">telefonia</field>
  <field name="cat">smartphone</field>
  <field name="features">iTunes, Telefonia, Accelerometro</field>
  <field name="features">Memorizza fino a 30,000 canzoni, 50,000
foto, o 300 ore di video</field>
  <field name="features">2.5-inch, 320x240 color TFT LCD display
with LED backlight</field>
  <field name="features">Fino a 20 ore di autonomia</field>
  <field name="features">Riproduce AAC, MP3, WAV, AIFF, Audible,
Apple Lossless, H.264 video</field>
  <field name="features">Blocco note, Calendario, Rubrica,
Messaggi, Data/Ora</field>
  <field name="includes">Cuffie, USB cable</field>
  <field name="weight">150</field>
  <field name="price">499.00</field>
  <field name="popularity">10</field>
  <field name="inStock">>true</field>
  <field name="manufacturedate_dt">2010-01-10T08:00:00Z</field>
</doc></add>
```

Salviamo il file nella cartella *example/exampledocs* (ad esempio come *iphone.xml*). Apriamo quindi il terminale, accediamo alla

cartella *example/exampledocs* e digitiamo

```
java -jar post.jar iphone.xml
```

per inserire i dati in formato XML utilizzando la procedura predefinita *post.jar*.

Di default, Solr crea una chiave primaria per il database chiamata *id*: nel caso in cui si inserisca un prodotto con un id già esistente, questo andrà a prendere il posto del vecchio prodotto.

Per cancellare un prodotto possiamo agire sempre da terminale specificando il valore di uno dei campi del prodotto che si vuole eliminare con il comando:

```
java -Ddata=args -jar post.jar  
"<delete><query>name:iphone</query></delete>"
```

Fatto questo, possiamo andare ad interrogare il nostro database direttamente dall'interfaccia amministrativa di Solr, nella sezione *Make a Query*, introducendo il testo da ricercare (ad es. *iphone*) e premendo il pulsante *Search*.

Le ricerche vengono eseguite utilizzando il metodo HTTP GET: è quindi possibile andare a inserire la propria query direttamente nell'URL di ricerca, assegnandola come valore alla variabile *q*. Ad esempio:

```
http://localhost:8983/solr/select/?q=iphone
```

restituirà tutti i dati dell'*iphone* appena inserito.

Se volessi inserire un criterio di ordinamento, mi basterebbe aggiungere alla query *&sort=price desc* per ordinare i risultati dal più caro al più economico.

Se volessi visualizzare solo alcuni campi dei risultati, mi basterebbe aggiungere alla query *&fl=name* per visualizzare solo il campo (*fl=field*) *name*.

I dati soddisfacenti i criteri di ricerca vengono restituiti in formato XML (fig.5).

```
<response>
  <lst name="responseHeader">
    <int name="status">0</int>
    <int name="QTime">1</int>
    <lst name="params">
      <str name="q">iphone</str>
    </lst>
  </lst>
  <result name="response" numFound="1" start="0">
    <doc>
      <arr name="cat">
        <str>elettronica</str>
        <str>telefonia</str>
        <str>smartphone</str>
      </arr>
      <arr name="features">
        <str>iTunes, Telefonia, Accelerometro</str>
        <str>Memorizza fino a 30,000 canzoni, 50,000 foto, o 300 ore di video</str>
        <str>2.5-inch, 320x240 color TFT LCD display with LED backlight</str>
        <str>Fino a 20 ore di autonomia</str>
        <str>Riproduce AAC, MP3, WAV, AIFF, Audible, Apple Lossless, H.264 video</str>
        <str>Blocco note, Calendario, Rubrica, Messaggi, Data/Ora</str>
      </arr>
      <str name="id">123456</str>
      <bool name="inStock">>true</bool>
      <str name="includes">Cuffie, USB cable</str>
      <str name="manu">Apple Computer Inc.</str>
      <date name="manufacturedate_dt">2010-01-10T08:00:00Z</date>
      <str name="name">Apple iPhone 32 GB</str>
      <int name="popularity">10</int>
      <float name="price">499.0</float>
      <float name="weight">150.0</float>
    </doc>
  </result>
</response>
```

fig. 5: Risultato della ricerca

Come abbiamo visto, i nuovi campi di ricerca possono essere dichiarati "al volo", ovvero semplicemente aggiungendo un elemento con il nome di un nuovo campo (ad es. `<field name="price">499</field>`). Tuttavia, per grandi moli di dati, è possibile definire prima una struttura mediante il file `schema.xml`. A partire dal file di esempio, si possono aggiungere in fondo al file, dopo una serie di comandi di inizializzazione, i nuovi campi. Ad esempio:

```
<field
  name="category" //creo un campo category
  type="text_ws" //di tipo testo
  indexed="true" //indicizzabile
  stored="true" /> //memorizzato nella struttura
```

RMS – RAPID MUSIC SEARCH

Descrizione

Al fine di testare il funzionamento di Solr, è stata realizzata una semplice applicazione PHP che utilizza Solr come motore di ricerca di dati in un database contenente l'elenco dei brani musicali in possesso di un appassionato di musica.

L'applicazione permette di indicizzare i brani caratterizzati da id, titolo, artisti, album, anno di uscita e di effettuare ricerche sui brani presenti, utilizzando le due caratteristiche principali di Solr: *highlighting* e *faceted search*.

Requisiti

I requisiti per utilizzare questa applicazione sono quattro:

- il server Solr, ovviamente;
- un web server con interprete PHP per poter usare l'applicazione in locale;
- un browser web;
- la libreria SolrPhpClient sviluppata da Donovan Jimenez e scaricabile dal sito <http://code.google.com/p/solr-php-client/>

Per questo esempio, utilizzeremo la versione 1.4.0 di Solr, EasyPHP come web server con interprete PHP nella sua versione 5.3.1.0, e Google Chrome 5 come browser con l'estensione XML Tree per la visualizzazione dei file XML (per debug).

Installazione

Come prima cosa, andiamo a applicare lo schema dei dati al server Solr, copiando il file `schema.xml` nella cartella `example/solr/conf`. In questo file sono riportati i campi del mio database, come vediamo in questa parte del codice. In particolare, vediamo che il campo *artist* è multivalore, in quanto un brano può essere cantato da più cantanti.

```
<fields>

  <field name="id" type="string" indexed="true" stored="true"
required="true" />
  <field name="title" type="text" indexed="true" stored="true"
required="true" />
  <field name="artist" type="text" indexed="true" stored="true"
required="true" multiValued="true" />
  <field name="album" type="text" indexed="true" stored="true" />
  <field name="year" type="text" indexed="true" stored="true" />

</fields>
<uniqueKey>id</uniqueKey>
<defaultSearchField>title</defaultSearchField>
```

Fatto questo, dobbiamo abilitare in Solr il supporto di PHP. Apriamo quindi il file `solrconfig.xml` sotto la cartella `example/solr` e decommentiamo le righe:

```
<queryResponseWriter name="php"
class="org.apache.solr.request.PHPResponseWriter"/>
<queryResponseWriter name="phps"
class="org.apache.solr.request.PHPSerializedResponseWriter"/>
```

Ora riavviamo il server Solr. Per testare il funzionamento, proviamo a fare una ricerca e ad aggiungere, all'URL, il parametro `"&wt=php"`. Se tutto funziona, i risultati della ricerca verranno strutturati in un array associativo (fig. 6), pronti per essere gestiti con PHP!

```
http://localhost:8983/solr... x
http://localhost:8983/solr/select/?q=Johnny+B+Good&version=2.2&start=0&rows=10&indent=on&wt=php

array(
  'responseHeader'=>array(
    'status'=>0,
    'QTime'=>1,
    'params'=>array(
      'indent'=>'on',
      'start'=>'0',
      'q'=>'Johnny B Good',
      'wt'=>'php',
      'version'=>'2.2',
      'rows'=>'10'),
  'response'=>array('numFound'=>1,'start'=>0,'docs'=>array(
    array(
      'id'=>array('4'),
      'title'=>array(
        'Johnny B Good'),
      'album'=>array(
        'Chuck Berry is on top'),
      'year'=>array(
        '1958'),
      'artist'=>array(
        'Chuck Berry'))
  )))
))
```

fig. 6: Risultati restituiti come array associativo

Ora copiamo i file dell'applicazione e la cartella SolrPhpClient dell'omonima libreria sotto la cartella www di EasyPHP, in modo che vengano indicizzate dal web server, e avviamo EasyPHP.

Apriamo il browser e visitiamo la pagina <http://localhost> per accedere ai file sul web server locale. Se tutto ha funzionato a dovere, dovrebbe apparire una pagina simile a quella di fig. 7.



RMS - Rapid Music Search

Questa semplice applicazione web consente di provare la potenza di Solr mediante l'utilizzo del linguaggio PHP per interfacciarsi a Solr che, a sua volta, si interfaccia con il database memorizzato sul computer locale. Il database è rappresenta una libreria musicale in cui ogni brano è caratterizzato da id, titolo, artisti, album e anno.

Scegli cosa vuoi fare:

- [Inserire un nuovo brano nell'indice](#)
- [Effettuare una ricerca e/o cancellare un brano dall'indice](#)

Questa applicazione web è stata realizzata da Davide Valeriani, matricola 190883, come progetto per gli esami di Ingegneria del Software A e di Reti di Calcolatori A tenuti dal prof. Agostino Poggi nell'a.a. 2009/2010.

fig. 7: Homepage dell'applicazione

Utilizzo

Le funzionalità offerte dall'applicazione sono sostanzialmente due:

- inserimento di nuovi brani nel database;
- ricerca e/o cancellazione dei brani nel database.

Le due pagine sono molto simili dal punto di vista dell'interfaccia grafica, per facilitare l'utilizzo da parte dell'utente.

Analizziamo le due macro-funzionalità singolarmente.

Inserimento

La pagina di inserimento si presenta con una semplice form (fig. 8) in cui è possibile inserire i dati del brano che si vuole inserire.

Interessante notare che, essendo *artisti* un campo multivalore, è richiesto di inserire gli artisti separati da virgole.



RMS - Rapid Music Search

INSERISCI BRANO

ID	<input type="text"/>
TITOLO	<input type="text"/>
ARTISTI	<input type="text"/> (separati da virgole)
ALBUM	<input type="text"/>
ANNO	<input type="text"/>

[Indietro](#)

Questa applicazione web è stata realizzata da Davide Valeriani, matricola 190883, come progetto per gli esami di Ingegneria del Software A e di Reti di Calcolatori A tenuti dal prof. Agostino Poggi nell'a.a. 2009/2010.

fig. 8: Pagina di inserimento brani

Una volta inseriti tutti i dati, è sufficiente premere il pulsante *Inserisci* per inserire il brano all'interno dell'indice del server Solr.

Ogni volta che si modifica l'indice, questo viene aggiornato e ottimizzato mediante due metodi appositi della libreria

SolrPhpClient: `$solr->commit()` e `$solr->optimize()`.

Se inseriamo un brano con un id già presente, questo andrà a sostituire il brano precedente. In questo modo è possibile anche modificare brani esistenti.

Questa funzionalità è possibile grazie al metodo

```
$solr->addDocuments($documents);
```

dove `$documents` è un array che contiene i dati da inserire.

Ricerca

La pagina di ricerca (fig. 9) permette all'utente di ricercare un brano fornendo alcune informazioni, inserite in un form simile a quello della pagina di inserimento. La ricerca viene effettuata in OR sul singolo campo e in AND tra i diversi campi. Questo ci permette di effettuare ricerche fornendo solo alcune parole nel caso di campi multipli.



RMS - Rapid Music Search

RICERCA

ID	<input type="text"/>
TITOLO	<input type="text"/>
ARTISTA	<input type="text"/>
ALBUM	<input type="text"/>
ANNO	<input type="text"/>
EVIDENZIA RISULTATI	<input type="checkbox"/>
RICERCA SFACCETTATA	<input type="checkbox"/>
<input type="button" value="Cerca"/>	

[Indietro](#)

Questa applicazione web è stata realizzata da Davide Valeriani, matricola 190883, come progetto per gli esami di Ingegneria del Software A e di Reti di Calcolatori A tenuti dal prof. Agostino Poggi nell'a.a. 2009/2010.

fig. 9: Pagina di ricerca

Tutto questo è reso possibile attraverso il metodo

```
$response = $solr->search($query, $offset, $limit);
```

in cui indichiamo la query da eseguire, il primo risultato che si vuole inserire nella risposta e quanti risultati, al massimo, possono essere ricavati. I risultati sono accessibili dall'oggetto:

```
$response->response->docs
```

Proviamo a cercare le canzoni di Ligabue presenti nella raccolta.

Scriviamo *Ligabue* nella casella *Artista* e premiamo il tasto *Cerca*.

Appare quasi istantaneamente la pagina con i risultati della ricerca, in un formato simile a quello di fig. 10.

A fianco di ogni risultato appare il link per poter cancellare il brano

dall'indice. Viene richiesta una conferma prima della cancellazione effettiva, per evitare cancellazioni accidentali.



RMS - Rapid Music Search

Documenti corrispondenti ai criteri di ricerca:

ID	TITOLO	ARTISTA	ALBUM	ANNO	
2	Il centro del mondo	Luciano Ligabue	Secondo tempo	2009	Cancella
3	Un colpo all'anima	Luciano Ligabue	Arrivederci, mostro!	2010	Cancella
8	Ho perso le parole	Luciano Ligabue	Radiofreccia	1998	Cancella
34	Caro mio Francesco	Luciano Ligabue	Arrivederci, mostro!	2010	Cancella
21	Il mio nome è mai più	Luciano Ligabue Jovanotti Piero Pelu	Il mio nome è mai più	2001	Cancella

[Nuova ricerca](#)

fig. 10: Risultati della ricerca

Proviamo ora a cliccare su *Nuova ricerca* per tornare alla pagina di ricerca. Notiamo che, sotto i campi di ricerca, è presente una checkbox *Evidenzia risultati* che permette di visualizzare i risultati della ricerca mettendo in evidenza le parole che soddisfano i criteri di ricerca. Proviamo quindi cercare nuovamente le canzoni di Ligabue, selezionando questa volta la casella di ricerca. Il risultato sarà una pagina simile a quella di fig. 11 in cui le parole soddisfacenti i criteri di ricerca vengono scritte in corsivo. Questa funzionalità di Solr si chiama *highlight*.



RMS - Rapid Music Search

Documenti corrispondenti ai criteri di ricerca:

ID	TITOLO	ARTISTA	ALBUM	ANNO	
2	Il centro del mondo	Luciano Ligabue	Secondo tempo	2009	Cancella
3	Un colpo all'anima	Luciano Ligabue	Arrivederci, mostro!	2010	Cancella
8	Ho perso le parole	Luciano Ligabue	Radiofreccia	1998	Cancella
34	Caro mio Francesco	Luciano Ligabue	Arrivederci, mostro!	2010	Cancella
21	Il mio nome è mai più	Luciano Ligabue Jovanotti Piero Pehu	Il mio nome è mai più	2001	Cancella

[Nuova ricerca](#)

fig. 11: Risultati della ricerca con funzione highlight

Per ottenere questo risultato, è sufficiente passare due parametri alla funzione di ricerca di Solr:

- hl = true, che attiva la funzione highlight
- hl.fl = *, per selezionare per quali campi attivare la funzione (la wildcard indica, ovviamente, di attivare la funzionalità per tutti i campi)

Creiamo quindi un array \$params in PHP in questo modo:

```
$params = array('hl' => 'true', 'hl.fl' => '*');
```

e chiamiamo il metodo *search* in questo modo:

```
$response = $solr->search($query, $offset, $limit, $params);
```

A questo punto, sarà sufficiente andare a recuperare i dati evidenziati dall'oggetto *\$response->highlighting*.

Un'altra funzionalità di Solr messa in luce da questa applicazione è la *faceted search* (ricerca sfaccettata) che consente di applicare

degli ulteriori filtri alla ricerca. Nel nostro esempio, useremo la ricerca sfaccettata per permettere all'utente di restringere ulteriormente la ricerca selezionando l'anno di uscita del brano musicale. Da notare come la ricerca sfaccettata restituisca preliminarmente il numero di brani risultanti dall'applicazione del filtro. Il risultato della ricerca sfaccettata lo vediamo nella fig. 12.



RMS - Rapid Music Search

Documenti corrispondenti ai criteri di ricerca:

ID	TITOLO	ARTISTA	ALBUM	ANNO	
2	Il centro del mondo	Luciano Ligabue	Secondo tempo	2009	Cancella
3	Un colpo all'anima	Luciano Ligabue	Arrivederci, mostro!	2010	Cancella
8	Ho perso le parole	Luciano Ligabue	Radiofreccia	1998	Cancella
34	Caro mio Francesco	Luciano Ligabue	Arrivederci, mostro!	2010	Cancella
21	Il mio nome è mai più	Luciano Ligabue Jovanotti Piero Pelu	Il mio nome è mai più	2001	Cancella

Statistiche per anno

Anno	N° documenti
2010	2
1998	1
2001	1
2009	1

[Nuova ricerca](#)

fig. 12: Esempio di ricerca sfaccettata

Per fare tutto questo è sufficiente aggiungere due ulteriori parametri all'array discusso in precedenza, che viene poi passato al metodo `$solr->search`:

- facet = true, che attiva la funzione di ricerca sfaccettata
- facet.field = year, per selezionare per quali campi attivare la funzione (il campo *year* in questo caso)

Creiamo quindi un array `$params` in PHP in questo modo:

```
$params = array('facet' => 'true', 'facet.field' => 'year');
```

e chiamiamo il metodo *search* in questo modo:

```
$response = $solr->search($query, $offset, $limit, $params);
```

A questo punto, sarà sufficiente andare a recuperare i dati evidenziati dall'oggetto `$response->facet_counts` mediante l'istruzione:

```
foreach ($response->facet_counts->facet_fields->year as  
$anno=>$numero)
```

che mi scorre i risultati dati da anno e numero di documenti soddisfacenti le condizioni e il filtro dell'anno.

CONCLUSIONI

Abbiamo visto le principali caratteristiche di Solr, la sua integrazione con PHP e un'applicazione di esempio che, ovviamente, potrebbe essere ulteriormente ampliata e arricchita di funzioni.

Solr, tuttavia, si integra anche con CMS (Content Management System, come Drupal), Ruby on Rails, Java e molti altri linguaggi di programmazione e, per questo, è molto utilizzato nei grandi siti web.

BIBLIOGRAFIA

- [Solr 1.4 Enterprise Search Server – D. Smiley, E. Pugh – Packt Publishing](#)
- <http://lucene.apache.org/solr/>
- http://www.lucidimagination.com/Community/Hear-from-the-Experts/Articles/Faceted-Search-Solr#faceting_impl
- <http://code.google.com/p/solr-php-client/>
- <http://www.ibm.com/developerworks/opensource/library/os-php-apachesolr/>
- <http://www.easyphp.org/>