

Lucene in Action

Java™ Application
Search Engine

java.sun.com/javaone/sf

Erik Hatcher

Coder / Writer / Speaker
eHatcher Solutions

www.ehatchersolutions.com



Goal

Empower you to be immediately effective at integrating Lucene into your applications.

Agenda

Demo

Meet Lucene

Case Studies

Indexing

Searching

Analysis

QueryParser

Document Handling

Demo

Searching with Lucene



Doug Cutting

The creator

- Information retrieval expert
- Xerox PARC: publications and patents
- Apple's Advanced Technology Group: V-Twin
- Excite
- Grand Central
- Nutch

Introduction

The project

- Apache Jakarta
 - 100% Pure Java™ initiative
 - With ports to other languages available
- Features
 - Interoperable index format
 - Fast and resource efficient algorithms
- History
 - 0.1 (Mar. 2000): First open source release
 - 1.3 (Dec. 2003): Compound index format
 - 1.4 (Spring 2004): Sorting, spans, term vectors

Lucene in Use

Powered by Lucene

- Jive Forums
- BobDylan.com lyrics search
- SearchBlox
 - Intranet search engine solution
- LingPipe
 - Open source
 - Government intelligence analysis
 - Biomedical research
- XtraMind: XM-InformationMinder™
 - Competitive and product intelligence
- Simpy: www.simpy.com

Case Studies

Lucene can handle it

- Used extensively in many domains
- Scales

Case Studies–Nutch

The NPR of search engines

- Alternative to Google
- Goals
 - Fetch billions of pages per month
 - Maintain index of those pages
 - Handle up to 1000 searches per second
 - High quality search results
 - Transparent ranking computation
 - Operate at minimal cost
- Public Nutch servers
 - mozdex.com, Yahoo! Labs, Objects Search

Case Studies–jGuru

Java™ technology FAQ and forum site

- 2,000,000 page views / month
- Indexes content, not site
- Features
 - Highly Java context-sensitive
 - Java keywords required in posts
 - Topic relevance checked
 - Indexes code (<pre>)
 - Multiple index searching

Lucene Index

The theory

- Concepts
 - Index: sequence of documents (a.k.a. Directory)
 - Document: sequence of fields
 - Field: named sequence of terms
 - Term: a *text* string (e.g., a word)
- Statistics
 - Term frequencies and positions

Indexing

Using IndexWriter

```
IndexWriter writer =  
    new IndexWriter(directory, analyzer, true);  
  
Document doc = new Document();  
  
    // add fields to document (next slide)  
  
writer.addDocument(doc);  
  
writer.close();
```

Indexing

Fields

```
doc.add(Field.Keyword("isbn", isbn));
doc.add(Field.Keyword("category", category));
doc.add(Field.Text("title", title));
doc.add(Field.Text("author", author));
doc.add(Field.UnIndexed("url", url));
doc.add(
    Field.UnStored("subjects", subjects, true));
doc.add(Field.Keyword("pubmonth", pubmonth));

doc.add(Field.UnStored("contents",
    author + " " + subjects));
doc.add(Field.Keyword("modified",
    DateField.timeToString(file.lastModified())));
```

Indexing

Field details

- Attributes
 - Stored: original content retrievable
 - Indexed: inverted, searchable
 - Tokenized: analyzed, split into tokens
- Factory methods
 - Keyword: stored and indexed as single term
 - Text: indexed, tokenized, and stored if String
 - UnIndexed: stored
 - UnStored: indexed, tokenized
- Terms are what matter for searching

Searching

Using IndexSearcher

```
IndexSearcher searcher =  
    new IndexSearcher(directory);
```

```
Query query = QueryParser.parse(  
    queryExpression,  
    "contents",  
    analyzer);
```

```
Hits hits = searcher.search(query);
```

```
for (int i = 0; i < hits.length(); i++) {  
    Document doc = hits.doc(i);  
    System.out.println(doc.get("title"));  
}
```

Searching

Query types

- TermQuery
 - Find by key
- RangeQuery
 - Text, date or numeric ranges
- BooleanQuery
 - Combine queries into expressions
- Others
 - PrefixQuery, WildcardQuery, FuzzyQuery...
- QueryParser
 - Turns readable expression into Query instance

Scoring

What is important

- Measure of document's relevance to a query
- Factors
 - tf: factor of term frequency in document
 - idf: factor of documents with term in index
 - boost: field-level boost
 - coord: factor-based # of query terms in document
 - queryNorm: normalization for query weights

Analysis

Processing terms into text

- Analysis occurs
 - For each tokenized field during indexing
 - For each term or phrase in QueryParser
- Several analyzers built-in
 - Many more in the sandbox
 - Straightforward to create your own
- Choosing the right analyzer is important!

Analyzing the Analyzer

Example phrase

The quick brown fox jumps over the lazy dog.

WhitespaceAnalyzer

Simplest built-in analyzer

The quick brown fox jumps over the lazy dog.

[The] [quick] [brown] [fox] [jumps] [over] [the]
[lazy] [dog.]

SimpleAnalyzer

Lowercases, splits at non-letter boundaries

the quick brown fox jumps over the lazy dog.

[the] [quick] [brown] [fox] [jumps] [over] [the]
[lazy] [dog]

StopAnalyzer

Lowercases and removes *stop* words

The quick brown fox jumps over the lazy dog.

[quick] [brown] [fox] [jumps] [over] [lazy] [dog]

SnowballAnalyzer

Stemming algorithm

The quick brown fox jumps over the lazy dog.

```
[the] [quick] [brown] [fox] [jump] [over] [the]  
[lazy] [dog]
```

Choosing an Analyzer

Considerations

- Stop word removal
 - Pros: smaller index
 - Cons: lose precision
- Stemming
 - jump, jumps, jumped, jumping
- Language
 - Stemming rules and stop words vary
 - CJK and other “symbolic” characters

QueryParser

Dealing with human-entered queries

- Parses query expression into Query instance
- Examples
 - “lucene in action”
 - +lucene publisher:manning
 - publishdate:[1/1/04 TO 12/31/04] subject:search
 - (search AND java) OR lucene

Parsing Documents

Extracting the text

- XML
 - SAX, DOM, Digester
- HTML
 - JTidy, NekoHTML, HTMLParser
- Office documents: Word, PDF, RTF, Excel...
 - POI, PDFBox, TextMining
 - Swing components
 - Open Office SDK

Summary

Why Lucene is important

- Search is important. Very important!
- Lucene is a fast, flexible Java application search engine
- Easy to integrate

For More Information

www.google.com–Search for “Lucene” :)

- Official site:
 - <http://jakarta.apache.org/lucene>
- Sandbox
 - jakarta-lucene-sandbox CVS repository
- Articles
 - java.net
 - Intro to Lucene
 - QueryParser Rules
- The book
 - *Lucene in Action* (Manning Publications)
 - Co-authored with Otis Gospodnetic

Q&A



Lucene in Action

Java™ Application
Search Engine

java.sun.com/javaone/sf

Erik Hatcher

Coder / Writer / Speaker
eHatcher Solutions

www.ehatchersolutions.com

