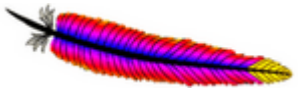






# Index-basierte Suche mit PyLucene

Thomas Koch

PyCologne, 14.07.2010

# Agenda

- Einführung  Index-basierte Suche
  - Motivation und Begrifflichkeiten
- Was ist PyLucene
  - Features
  - Historie
- Anwendung von PyLucene
  - Index erstellen
  - Suchanfragen
- Demonstration von PyLucene
  - Hands-on & Code-Beispiele
- Weiterführende Themen
  - Links, Literatur, Related work

# Einführung: Index-basierte Suche

Grundlagen Index-basierter Suche:

- Index verwaltet 'Dokumente' (abstraktes Objekt)
- pro Dokument: versch. Metadaten (i.W. key-value)
- Inhalte werden im (invertierten) Index verwaltet (optimiert für Suchanfragen)
- Indexer benötigen **Text**-Darstellung!
- Suche: Anfrage über Felder und Werte (sog. „Terme“)
- Abfrage über Index liefert Ergebnisse ("Hits")

# Herausforderungen bei Indexierung

- Indexierung
  - Daten/Dateistrukturen (für Index)
  - Textanalyse (Stemmer, Tokenizer …)
- Suche
  - Abfragesprache (Query Language)
  - Matching-Algorithmen, Ranking (Scores) etc.
- Allgemein
  - Mehrsprachigkeit (Encodings, Stop-Wörter, Plural …)
  - Extraktion der Daten (z.B. Web-Crawler)
  - Konvertierung (Dokument-Formate)
  - **Performance!!!**

# Lösung: Frameworks für Suchmaschinen

- Einfaches Muster:
  - 1. **Indexierung** : Dokumente → TextAnalyse → Index
  - 2. **Suche**: Anfrage → Parsing / Query → Resultat
  - 3. **Visualisierung** der Suchergebnisse (Anwendungsabhängig)
- Unterstützung durch Framework: Java Lucene
  - *Lucene is a high-performance, full-featured **text search engine written** entirely in **Java**. It is a technology suitable for nearly any application that requires full-text search, …*
  - API (Auszug…)

| | |
|---|---|
| org.apache.lucene.**analysis** | *convert text into indexable/searchable tokens.* |
| org.apache.lucene.document | *logical representation of a Document* |
| org.apache.lucene.**index** | *maintain and access indices* |
| org.apache.lucene.queryParser | *query parser* |
| org.apache.lucene.**search** | *search indices* |
| org.apache.lucene.store | *Binary I/O API, used for all index data* |
| org.apache.lucene.util | *utility classes* |

# Was ist PyLucene?

- About
  - PyLucene ist eine Python Erweiterung (package 'lucene')
  - PyLucene erlaubt Zugriff auf alle Lucene Features von Python
- Implementierung:
  - PyLucene ist keine Portierung nach Python, sondern „Wrapper“ um Java Lucene!
    - *PyLucene embeds a Java VM with Lucene into a Python process.*
  - Python Module sind durch JCC generiert (aus Java-Sourcen):
    - *PyLucene is built with JCC, a C++ code generator that makes it possible to call into Java classes from Python via Java's Native Invocation Interface (JNI).*
- Aktuelle Version(en)
  - PyLucene 3.0.2-1 and 2.9.3-1 (Juli 2010)
    - JavaLucene 2.9.3: Java 2.x series, based on Java 1.4.
    - JavaLucene 3.0.2: Java 3.x series, based on Java 5.

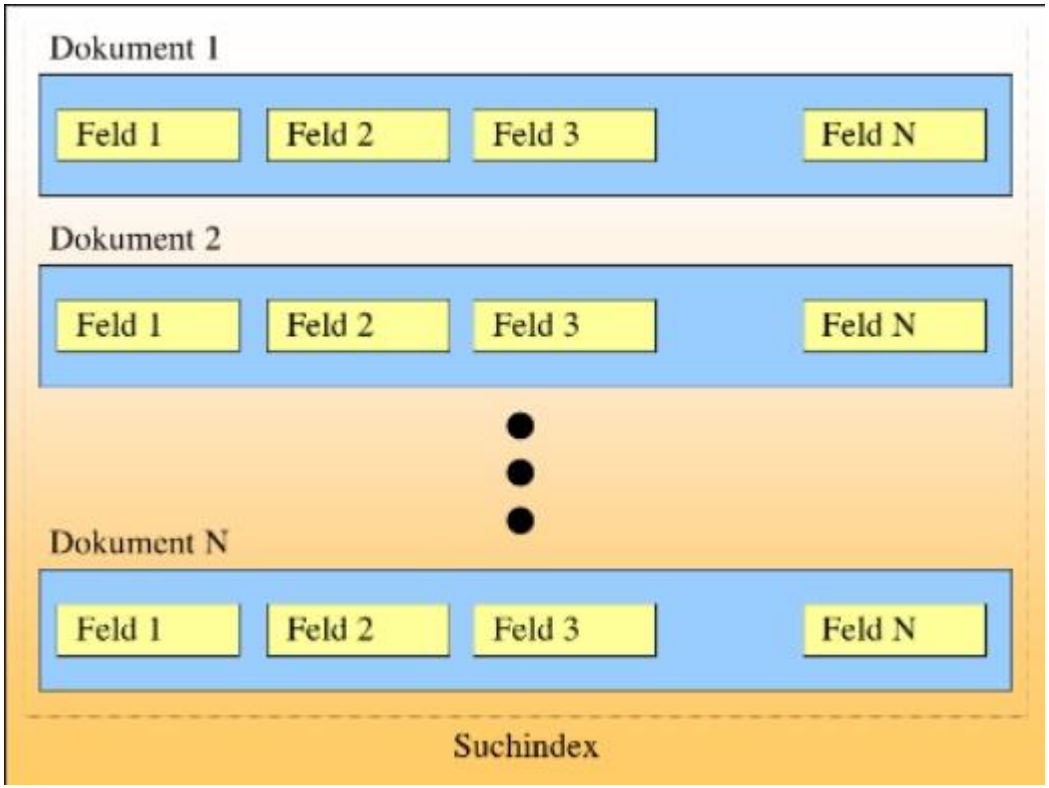
# Was ist PyLucene?

- Features
  - Vollständige Abdeckung von Java Lucene in Python
  - API fast 1:1 kompatibel (zu Java-Lucene)
- Abhängigkeiten
  - Runtime: Erfordert Python 2.x (x>=3) und Java 1.4+
  - Build: Erfordert make, ant, C++ compiler
  - Läuft auf Mac OS X, Linux, Solaris und Windows
- Historie
  - Lucene 1.0: entwickelt von Doug Cutting (2000) http://cutting.wordpress.com/
  - PyLucene : entwickelt von Andi Vajda
    - Zunächst *GCJ-compiled version of Java Lucene integrated with Python (~2004)*
    - Später als *JCC-generated Python wrapper around Java Lucene (2007)* http://blog.chandlerproject.org/author/vajda/
    - Seit V2.4 is PyLucene *subproject of the Apache Lucene project (2009)* http://lucene.apache.org/pylucene/

# Anwendung von PyLucene

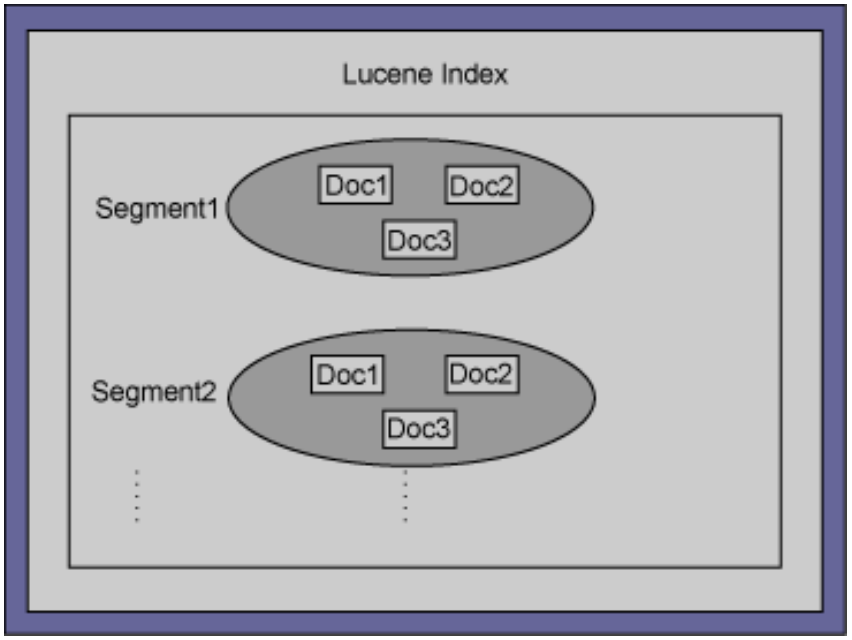
- Übersicht der wichtigsten Module
  - Document
    - Einheit der Indexierung; i.W. Menge von Feldern (Field)
  - Field
    - Teil eines Document; Name, Value; Properties: indexed, stored
  - Term
    - Wort in einem Document; wird zur Suche verwendet
  - IndexWriter
    - Index anlegen und erweitern
  - IndexReader
    - Nur lesender Zugriff, oder: Index ändern (delete Document)
  - IndexSearcher
    - Suche durchführen
  - Analyser
    - Textanalyse: extrahiert Terme aus einem Text

# Dokumente und Felder




http://it-republik.de/zonen/magazine/onlineartikel/pspic/picture_file/94/Bild14a168ae122fb6.jpg

# Index : Segmente und Dokumente




http://www.ibm.com/developerworks/opensource/library/wa-lucene/logic_view.gif

# Anwendung von PyLucene: Indexierung

- Index anlegen

```
analyzer = lucene.StandardAnalyzer()
store  = lucene.SimpleFSDirectory(lucene.File(storeDir))
writer = lucene.IndexWriter(store, analyzer, True,  # create
             lucene.IndexWriter.MaxFieldLength.LIMITED)
```

- Index erstellen

```
for file in files:
  doc = lucene.Document()
  # store path info for later use
  doc.add(lucene.Field("path", get_path(file),
                       lucene.Field.Store.YES,
                       lucene.Field.Index.NOT_ANALYZED))
  contents = file.readlines()
  doc.add(lucene.Field("contents", contents,
                       lucene.Field.Store.NO,
                       lucene.Field.Index.ANALYZED))
  # index document:
  writer.addDocument(doc)
```

# Anwendung von PyLucene: Suche

- Suchanfrage vorbereiten

```
directory = SimpleFSDirectory(File(storeDir))
searcher = IndexSearcher(directory, True) # readOnly
analyzer = StandardAnalyzer() # same as for indexing
```

- Suchanfrage durchführen

```
query = QueryParser("contents", analyzer).parse(queryStr)
# perform query on index and get max=50 results
scoreDocs = searcher.search(query, 50).scoreDocs
# finally show results
for scoreDoc in scoreDocs:
  doc = searcher.doc(scoreDoc.doc) # get Document from index
  print 'match:', doc.get("path")
```

# Anwendung von PyLucene

- PyLucene Initialisierung
  - vor dem Aufruf von PyLucene muss JVM geladen werden

    >> import lucene

    >> lucene.initVM(classpath=lucene.CLASSPATH)
    - Angabe weiterer VM-Parameter möglich: initialheap, vmargs etc.
- PyLucene Konfiguration und Optimierung
  - writer.setMaxBufferedDocs(int)
    - *minimal number of documents before the buffered documents are flushed*
  - writer.setMaxFieldLength(int)
    - *maximum number of terms per field*
  - writer.setXYZ ()
    - MergePolicy, MaxMergeDocs,MergeFactor …
  - writer.optimize()
    - Index-Optimierung anstossen (Synchron oder asynchron)
  - writer.close()
    - Auch auf reader und searcher immer close() aufrufen!

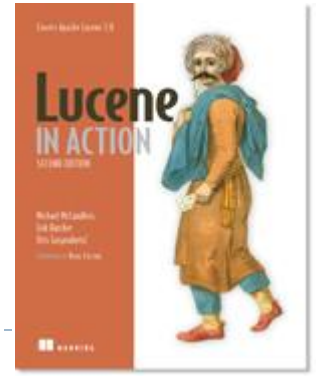
# Code-Beispiel

Demonstration

Thomas Koch 7/2010

# Weiterführende Themen / Infos

- Specials
  - Lucene-Features: Filter, Scorer, Explanation, Boosting etc.
  - Verschiedene/eigene Analyzer: **GermanAnalyzer**
  - Verschiedene/eigene Query-Methoden: **FuzzyQuery**
  - PyLucene in Python erweitern: PorterStemmerAnalyzer
  - Build JCC/PyLucene from source…
- Related
  - Apache Solr http://lucene.apache.org/solr/
  - *A powerful full-text search, hit highlighting, faceted search, dynamic clustering, database integration, and rich document (e.g., Word, PDF) handling.*
- Literatur
  - Lucene in Action, Manning Pub. (2nd Edition)
    **Beispiel-Code aus LIA-Buch in Distribution! (/samples)**
    http://www.manning.com/hatcher2/
  - Dr. Christian Herta „Indizierung mit Lucene“
    **Suchtechnologie und Information Retrieval**
    http://www.christianherta.de/



# Links

- Sample Data:
  - Enron Email Dataset
    http://www.cs.cmu.edu/~enron/
  - 517k files
  - 1.3GB text
- Lucene Index Viewer
  - **Luke** - Lucene Index Toolbox
    http://code.google.com/p/luke/
- Apache PyLucene
  - project home page: http://lucene.apache.org/pylucene
  - mailing list: pylucene-dev@lucene.apache.org
  - svn repo: http://svn.apache.org/repos/asf/lucene/pylucene/

# Breaking News:
# JCC and PyLucene ported to Python 3.0

Am 12.07.2010, 21:38 Uhr, schrieb Andi Vajda <vajda@apache.org>:

> Today, I ported JCC and PyLucene (trunk) to **Python 3.1.2**
> on Mac OS X 10.6.
> All unit tests and Lucene in Action samples pass. **2to3 did wonders**. So
> did the docs for all the manual C code changes.
>
> If you'd like to check it out:
> - install Python 3.1.2
> - install the latest distribute
> - check out
> http://svn.apache.org/repos/asf/lucene/pylucene/branches/python_3/
>
> Bugs, comments, etc.. welcome !
>
> Andi..