>A Lucene-based cataloging solution for Zope-3 and CPS-3.4

EuroPython 2006, Geneva

Julien Anguenot <ja@nuxeo.com>

Créat. : 05/07/2006
Modif. : 05/07/2006

Who am I ?

- Julien Anguenot, « core » Nuxeo R&D architect
- Nuxeo is editing with the help of a community of contributors
  - Nuxeo CPS (GPL / ZPL)
    - Open Source ECM
    - http://www.cps-project.org
  - Apogée (EL)
    - Rich client for ECM applications based on Eclipse RCP
    - Project of the Eclipse foundation
    - http://apogee.nuxeo.org
- Nuxeo is also a « core » contributor of several open souce projects including those listed below
  - Zope / Zope3 / CMF : http://www.zope.org
  - Z3/ECM : http://www.z3lab.org
  - ZODB : http://dev.zope.org/ZODB/

Outline

- Motivations
- CMF catalog / zope.app.catalog limitations
- Lucene
- PyLucene
- NXLucene
- nuxeo.lucene
- CPSLuceneCatalog
- Production feedbacks
- More information
- Q & A

Motivations are

- To have a scalable solution for searching and indexing with Zope-3, CPS-3.4 and CPS-4
  - It must scale with millions of documents !
  - Documents, in this case, are office documents indexed as fulltext along with metadata.
- To enhance Zope global performances
  - when the ZODB is populated with millions of objects...
  - when the Zope AS is heavily loaded (R/W)
- To have more powerful search capabilities
  - Rich and « easy to use » query language
  - Fulltext analysis (French, German, Synonyms, Stemming etc...)
  - Content tagging
  - Etc...

CMF catalog (ZCatalog based) limitations

- Performances start to decrease around 100k documents and get really bad around 300k
  - R/W
- Global Zope AS performances are decreasing as well around the same amount of documents.
  - R/W
- Zope does not necesarily need to deal with indexing and searching (In an ideal world ZCatalog wouldn't even exist)
- Batching is not possible ZCatalog side.
- ZCatalog code is part of the really early Zope code and thus is not really funny to maintain
- Zope3 catalog (zope.app.catalog) does almost nothing. (TXNG 3 is great though)

Furthermore...

- Several Open Source full-featured text search engine libraries are available, maintained by experts and globally superiors :
  - Xapian, Lucene, OpenFTS (pgsql), ...
- Thus let's save us some time and energy !

Lucene rocks ! (1/2)

- Full-featured text search engine library
- Open Source, Apache Software Foundation
- Java based
  - Cross platform
- Easy to use
- Flexible
- Powerful
- High-performance
- Powerful query language
- Model of good object-oriented architecture
  - Powerful abstraction

1932394281

Lucene rocks ! (2/2)

- Supports several backends
  - FS, BDB, RAM
- Several implementations available
  - Perl, Python, C++, .NET, Ruby
- Lots of interesting and concret projects based on Lucene
  - Nutch for instance.
  - Lots and lots of projects using it !
- Tools to manage Lucene stores do exist already
  - Luke for instance
- Good documentation including a complete tutorial (LIA)
- Lucene In Action book (ISBN : 1932394281)
- Current version : Lucene-2.0.0

PyLucene (1/2)

- Most recent Java Lucene port
- Part of the Chandler project at first
- Maintained by the OSAF (Open Source Application Foundation) => Andi Vajda
- Not a true port
  - Which is good
- Use GCJ and SWIG (used to before PyLucene 2.0) to export Java Lucene API to Python
  - Compile java code in a native shared library
  - Exposes C++ classes
  - C++ / Python integration is simple
- Same API as the Java Lucene one
  - Java Lucene documentation is accurate
- Indexes are compatible with the Java Lucene ones

PyLucene (2/2)

- Closest port of Java Lucene out there
  - Easy to upgrade
- Performances
  - Outperforms Java Lucene ! Eh eh !
- Python Python Python !
  - Lucene has a really nice object-oriented model
  - It plays really well with Python
  - Let the code speaks for itself (compare Java / Python code)

NXLucene

- Standalone indexation server handling Lucene stores
- Twisted based
- Multi-threaded
- PyLucene based
- Uses some parts of the Zope3 component architecture
  - zope.testing
  - zope.interface
- NXLucene is **not** running on top of the Zope AS !
- Under the LGPL

NXLucene : XML-RPC server interface

- twisted.web.xmlrpc server
- Supports HTTP/1.1 using Keep-Alive
  - See nuxeo.lucene and the HTTP persistent transport implementation.
  - Waiting for the twisted.web2 better HTTP/1.1 support
- Simple API
  - indexDocument(uid, xml_query='', b64=False, sync=False)
  - reindexDocument(uid, xml_query='', b64=False,sync=False)
  - unindexDocument(uid, sync=False)
  - clean()
  - optimize()
  - getNumberOfDocuments()
  - hasUID()
  - searchQuery(xml_query)
  - search(query_str)
- Result sets are returned as RSS streams

NXLucene : multi-threading

- PyLucene.PythonThread
  - A threading.Thread extension that delegates starting of the actual OS thread to libgcj. In order to keep libgcj's garbage collector happy, any python thread using libgcj must be of this class.
- Write operations are performed async by default
  - PyLucene.PythonThread flexible thread pool for writing operations
- Search operations are sync
  - Within the main thread
  - Or using twisted deferred (not on the trunk right now)
- Global write lock on the core server for now
  - Not a problem for us with CPS since only one index
  - Would be better to take advantage of the Lucene store locks. But more sensitive to implement

NXLucene : PyLucene based

- Uses PyLucene 1.9.1 on the trunk
  - PyLucene 2.0.0 planned for the 2.x branch
- FSDirectory backend only for now
  - Not transactional
  - Is Bdb really working IRL ? ;)
  - See ZODB after commit hook within nuxeo.lucene
- PyLucene extensions
  - Analyzers
    - French, Sort, etc...
  - Pre-configured fields
    - Path field
    - Keyword field
    - MultiKeyword field
    - Sort field
    - etc...

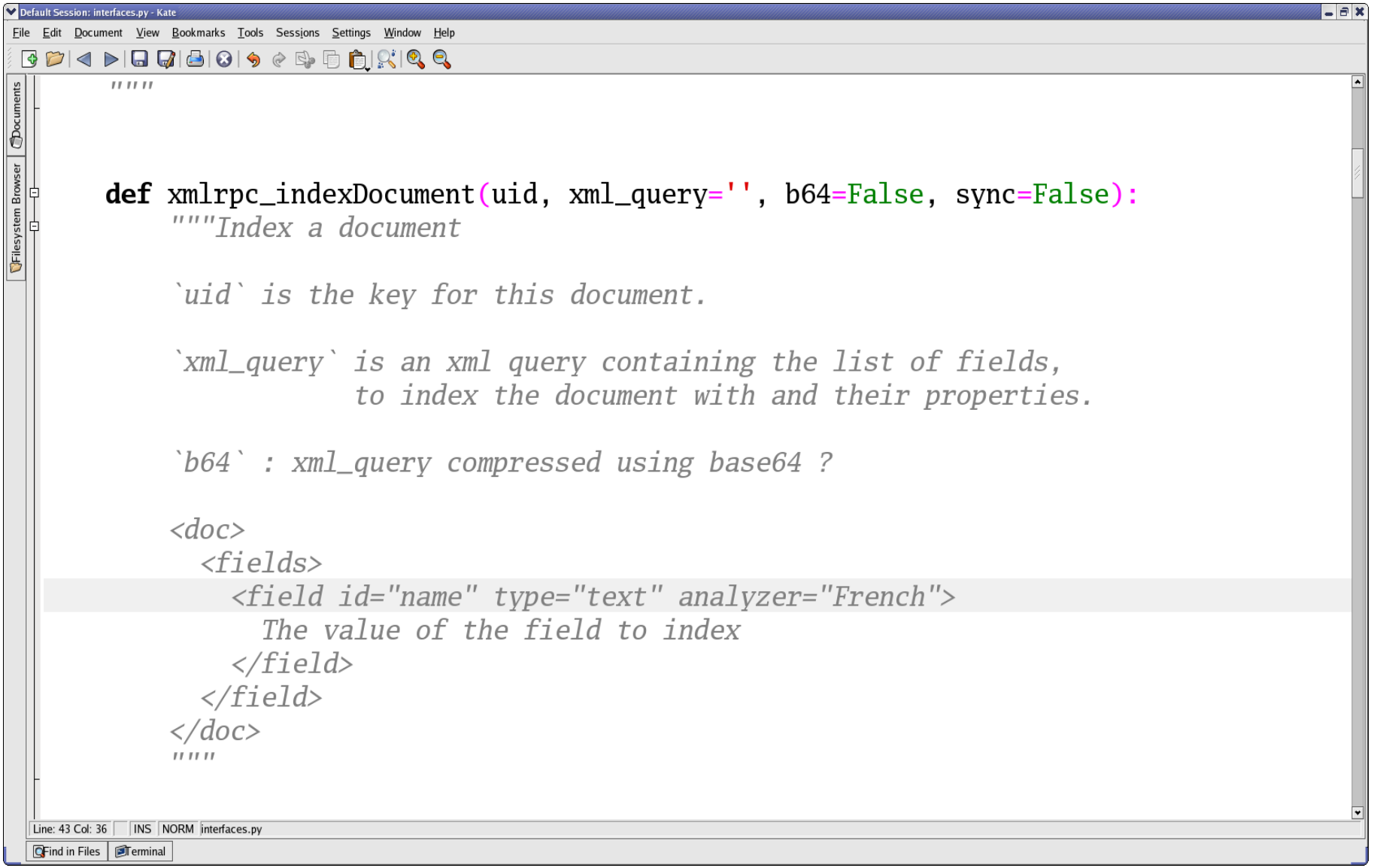
NXLucene : nxlucene core lib for Python programs

- When installing NXLucene you get :
  - The actual NXLucene server
  - nxlucene core lib
- nxlucene.testing
  - To ease testing Python applications requesting a NXLucene server using XML-RPC fake connections
- nxlucene.rss
  - RSS <-> Python adapters
- nxlucene.xmlquery
  - To produce XML queries for NXLucene

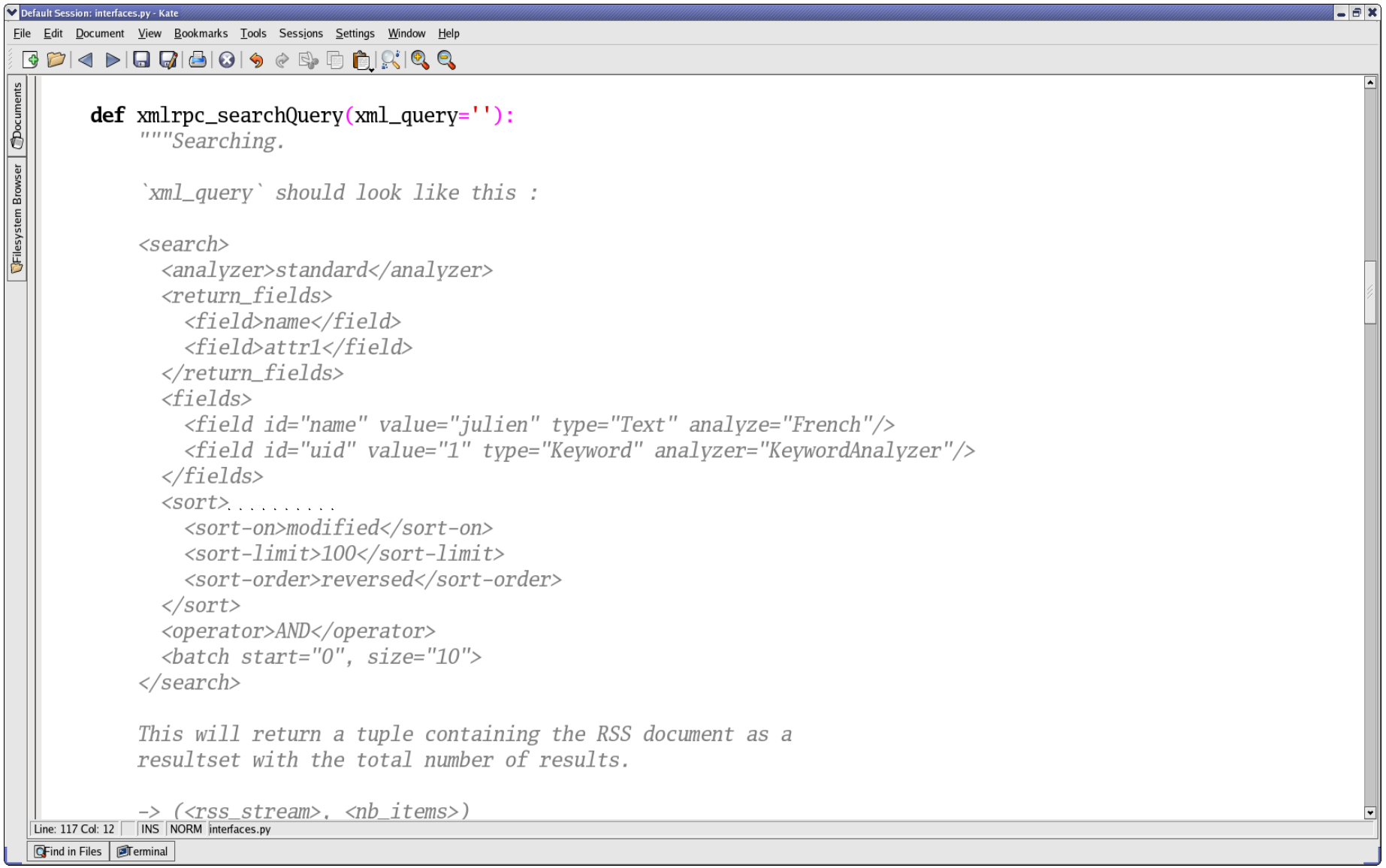
NXLucene : XML queries

- NXLucene goal :
  - Not Python specific for client applications !
  - Easily usable from whatever languages using XML and XML-RPC libraries (standard)
  - No internal Lucene knowledge should be mandatory for the client requesting NXLucene.
- Custom XML Query for both writing and searching
  - I'll allow the direct use of Lucene query string using simple GET in the future. (API's ready)
- Ability to express more complex queries including analyzers, field types, encoding, etc..
  - Lucene query strings are of course supported here on terms
- Searches return RSS streams
- Let's check some examples

NXLucene : example of XML query for indexation



NXLucene : example of XML search query



NXLucene : what's in the pipe ?

- Optimisations
  - In production with larger and larger indexes stores
  - Profiling
- Core refactoring
- PyLucene 2.0.0
- twisted.web2
- ICE connector
- Twisted Perspective Broker (PB) ?
- SOAP interface ?
- NXLucene extensions for analyzers, fields etc...
- Packaging
- Bring your use cases !

nuxeo.lucene (1/2)

- Python cataloging component using the Zope-3 component architecture
- Abstraction for Python objects indexation
  - Know how to index Python objects in NXLucene
  - Know how to retrieve documents from the store and generate back Python result sets from RSS
- XML-RPC proxy for NXLucene (IXMLRPCLuceneServer)
  - Note, here whatever XML-RPC remote server could be theoritically used. (think adapter)
  - Custom persistent transport to avoid creating one HTTP connection per request
  - Force HTTP/1.1. (see twisted.web limitations regarding HTTP/1.1)
- Use of ZODB after commit hook to deal with the fact that the Lucene FSDirectory backend is not transactional

nuxeo.lucene (2/2)

- Python objects discriminated using interfaces
- LuceneFieldConfiguration objects
  - Name (-> fullname)
  - Attribute (-> getFullname)
  - Type (-> Text)
  - Analyzer (-> French)
- Filtering out Lucene stored fields using a global result schema
- ZCML meta for fields and columns registration
- Let's see an example

nuxeo.lucene : example

- nuxeo.lucene.README.txt doctest

nuxeo.lucene status

- Only the Python « layer » of the component is implemented
  - Not fully integrated with Zope3 (no views, subscribers, etc...)
  - Because of Five limitations regarding local utilities (Zope-3.2 time)
- Contributors welcome :)
- Under the ZPL

CPSLuceneCatalog

- CMF portal_catalog complete replacement for CPS-3.4.
- Based on nuxeo.lucene (Zope3 component)
- Implements CMFCore.ICatalogTool
- BBB with ZCatalog queries
  - At least the subset CPS was using
  - Brains along with acquisition are returned
  - Extend CMFCore.ObjectObjectWrapper
- Add CPS specific business rules regarding indexation
  - Versionning
  - Filter sets
  - Etc...
- We still use CMF security
- GenericSetup support

First production feedbacks

- More than 600K documents for one of our customer nowadays (50 fields (including fulltext) per document and the store weights about 1Go optimized)
- Fast as expected !
  - We are still enhancing
- Reaching 1.x stability (NXLucene-0.9.x)
- Difficulties
  - Getting the right GCC on which compiling PyLucene on a given platform to avoid core dumps...
  - Use GCC-3.4.6 on Redhat / Fedora will save you some time
- NXLucene is not necessarly the best solution for metadata
  - Aggregating results from two backends for fulltext, on the one hand, and metadata, on the other hand, would be certainly better ?

More information

- Websites
  - NXLucene : http://www.cps-project.org/workspaces/development/projects
  - CPS Platform : http://www.cps-project.org
  - PyLucene : http://pylucene.osafoundation.org/
  - Lucene : http://lucene.apache.org/
  - Twisted : http://twistedmatrix.com/projects/core/
- Looking for support ?
  - http://lists.nuxeo.com/mailman/listinfo/mailman/cps-devel
- Looking for commercial support ?
  - http://www.nuxeo.com/en/

Q&A

Thank you for your attention !

Contact information

- Julien Anguenot, janguenot@nuxeo.com
  - Nuxeo SAS
  - 16, rue du Solleillet
  - 75020 Paris
  - France
  - contact@nuxeo.com