

Fun with Flexible Indexing

Mike McCandless, IBM

10/8/2010



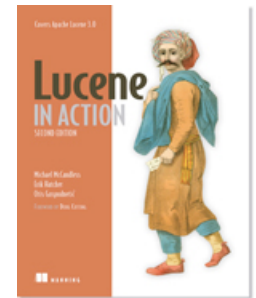
Agenda

- **Who am I?**
- **Motivation**
- **New flex APIs**
- **Codecs**
- **Wrap up**



Who am I?

- **Committer, PMC member Lucene/Solr**
- **Co-author of Lucene in Action, 2nd edition**
 - LUCENEREV40 promo code!
- **Blog: <http://chbits.blogspot.com>**
- **Emacs, Python lover**
- **Sponsored by IBM**



Your ideas will go further if you don't insist on going with them.

Motivation

- **Lucene is showing its age**
 - vInt is costly
- **Lucene is hard to change, at low-levels**
 - Index format is too rigid
- **Yet, lots of innovation in the IR world...**
 - New compression formats, data structures, scorings models, etc.
- **IR researchers use other search engines**
 - Terrier, Lemur/Indri, MG4J, etc.



Better to ask forgiveness than permission.

An example: omitTFAP

- Added in version 2.4
- Turns off positions, termFreq
- 50 KB patch, 25 core source files!
- Follow-on (LUCENE-2048) still open...
- This was a simple change!
 - What about harder changes, eg better encoding?
- Yes, devs can make these changes... but that's not good enough



Actions speak louder than words.

Motivation

- **Goal 1: make innovation easy(ier)**
 - You shouldn't have to be a rocket scientist to try out new ideas
 - But: can't lose performance
- **Goal 2: innovate**
 - Catch up to state-of-the-art in IR world



If you're not making mistakes, you're not trying hard enough.

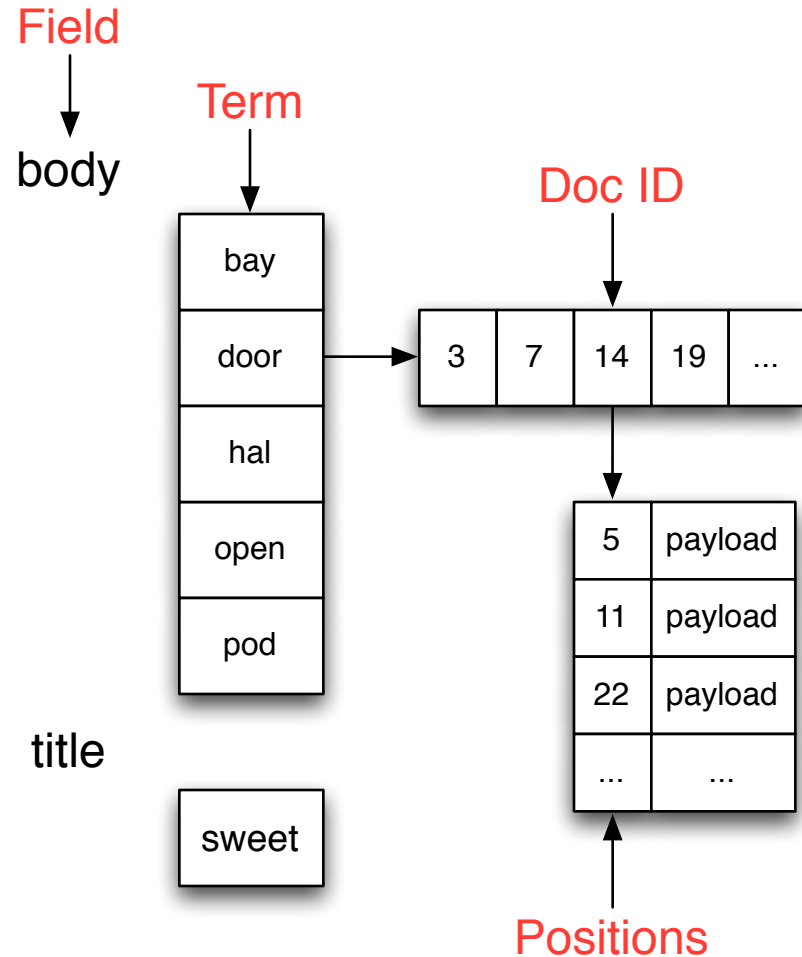
Agenda

- Who am I?
- Motivation
- **New flex APIs**
- Codecs
- Wrap up



Inverted Index 101

```
SortedMap<Field,  
  SortedMap<Term,  
    List<Doc ID,  
      List<Pos, Payload>  
    >  
  >  
>  
>
```



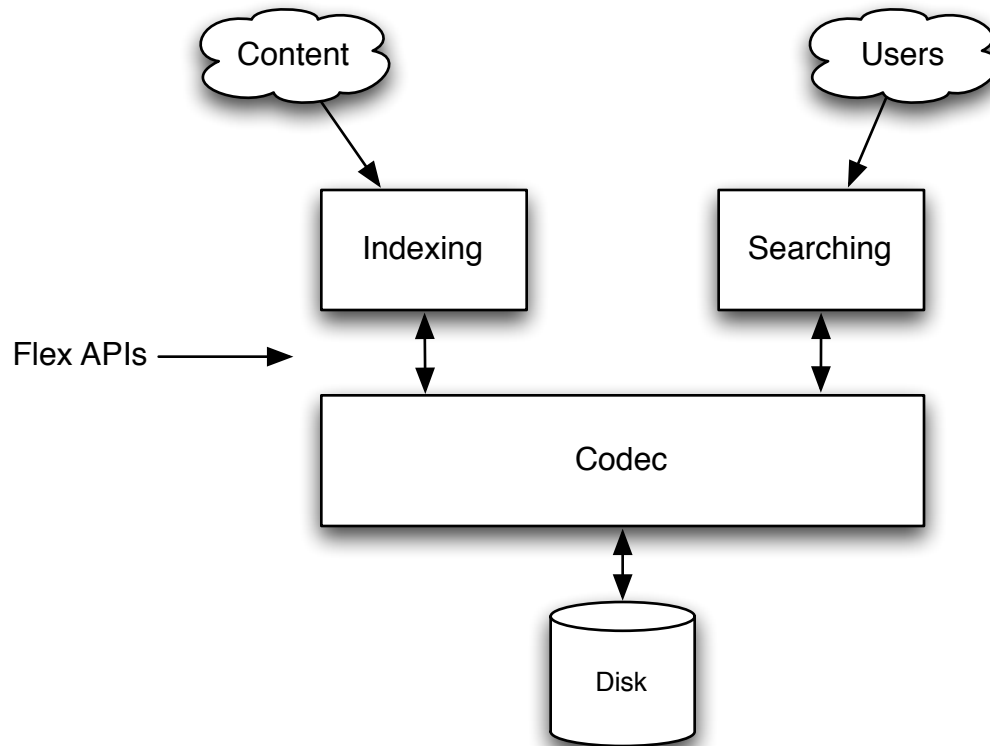
Flex overview

- **4.0 (trunk) only!**
- **New low-level postings enum API**
- **Pluggable, per-segment codec has full control over reading/writing postings**
 - Building blocks make it easy to create your own
 - Some neat codecs!
- **Performance gains**
 - Much less RAM used
 - Faster queries, filters



Don't trade your passion for glory.

Flex is very low level



4D enum API

- **Fields, FieldsEnum**
 - field
- **Terms, TermsEnum**
 - term, docFreq, ord
- **DocsEnum**
 - docID, freq
- **DocsAndPositionsEnum**
 - docID, freq, position, payload
- **All enums allow custom attrs**



If two people always agree, one is not necessary.

API: TermsEnum

- **Iterates through all unique terms**
 - Separates terms from field
- **Each term is opaque, fully binary**
 - BytesRef (slices a byte[])
 - New analysis attr provides BytesRef per token
 - Collation, numeric fields can use full term space
- **Char terms can use any encoding**
 - Default is UTF8 (some queries rely on this)
 - Others are possible (eg BOCU1, LUCENE-1799)



Absolute power corrupts absolutely.

API: TermsEnum

- You can now re-seek an existing TermsEnum
- Seek gives explicit return result
 - FOUND, NOT_FOUND, END
- Ord, seek-by-ord (optional, only for segment)
- Enables seek-intensive queries
 - Eg AutomatonQuery
 - FuzzyQuery is much faster for N=1,2!
 - New automaton spell-checker also uses FuzzyTermsEnum (LUCENE-2507)



Life is about the journey, not the destination.

API: TermsEnum

- **Term sort order is determined by codec**
 - `Comparator<BytesRef> getComparator()`
- **Core codecs use unsigned byte[] order**
 - Unicode code point if byte[] is UTF8
- **If you change this, some queries won't work!**



There is no security on this earth; only opportunity.

FieldCache improvements

- **FieldCache consumes the flex APIs**
- **Terms / terms index field cache more RAM efficient, low GC load**
 - Used with SortField.STRING
- **Shared byte[] blocks instead of separate String instances**
 - Term remain as byte[]
- **Packed ints for ords, addresses**
- **RAM reduction ~40-60%**



Happiness = expectations minus reality.

API: Docs/AndPositionsEnum

- **API very similar to 3.x**
 - Still extends DISI
- **TermsEnum provides Docs/AndPositionsEnum**
- **Bulk read API exists but still in flux (LUCENE-1410)**
- **You provide the skip docs**
 - Deleted docs are no longer silently skipped



The best way to learn is to do.

Custom skip docs

- **IndexReader provides .getDeletedDocs**
 - Replaces .isDeleted
- **Queries pass the deleted docs**
 - But you can customize!
- **Example: FilterIndexReader subclass**
 - Apply random-access filter “down low”
 - ~40-130% gain for many queries, 50% filter
 - LUCENE-1536 is the real fix
 - <http://s.apache.org/PNA>



Fish for someone, they eat for a day. Teach them to fish, they eat for a lifetime.

Agenda

- Who am I?
- Motivation
- New flex APIs
- **Codecs**
- Wrap up



What's really in a codec?

- **Codec provides read/write for one segment**
 - Unique name (String)
 - FieldsConsumer (for writing)
 - FieldsProducer is 4D enum API + close
- **CodecProvider creates Codec instance**
 - Passed to IndexWriter/Reader
- **You can override merging**
- **Reusable building blocks**
 - Terms dict + index, Postings



Sweet are the uses of adversity.

Testing Codecs

- All unit tests now randomly swap codecs
- If you hit a random test failure, please post to dev, including random seed
- Easily test your own codec!



Always under-promise and over-deliver.

lucid
IMAGINATION

Standard codec

- **Default codec**
 - On upgrade, newly written segments use this
- **Terms dict: PrefixCodedTerms**
- **Terms index: FixedGapTermsIndex**
- **Postings: StandardPostingsWriter/Reader**
 - Same vInt encoding as 3.x



Don't attribute to malice that which can be otherwise explained.

PrefixCodedTerms

- **Terms dict**
- **Responsible for Fields/Enum, Terms/Enum**
 - Maps term to byte[], docFreq, file offsets
- **Shared prefix of adjacent terms is trimmed**
- **Pluggable terms index, postings impl**
- **Format**
 - Separate sections per-field



Imagination is more important than knowledge.

FixedGapTermsIndex

- **Every Nth term is indexed**
 - Loaded fully into RAM
- **RAM image is written at indexing time**
 - Very fast reader init, low GC load
 - Parallel arrays instead of instance per term
- **Index term points to edge between terms**
 - Vs 3.x where index term was a full entry
- **Useless suffix removal**
 - a, abracadabra



The reasonable person adapts himself to the world...

FixedGapTermsIndex

- **Much better RAM/GC efficiency**
- **HathiTrust terms index**
 - 22.2 M indexed terms
 - 3.x: 3974 MB RAM, 72.8 sec to load
 - 4.0: 401 MB RAM, 2.2 sec to load
 - 9.9 X less RAM, 33X faster
- **Wikipedia 3.8X less RAM**
 - <http://s.apache.org/OWK>
- **Default terms index gap changed 128 -> 32**



...the unreasonable one persists in trying to adapt the world to himself...

PreFlex codec

- Reads 3.x index format
- Read-only!
 - Except: tests swap in a read/write version
- **Surrogates dance dynamically reorders UTF16 sort order to unicode**
 - Sophisticated backwards compatibility layer!



..therefore all progress depends on the unreasonable person.

Pulsing codec

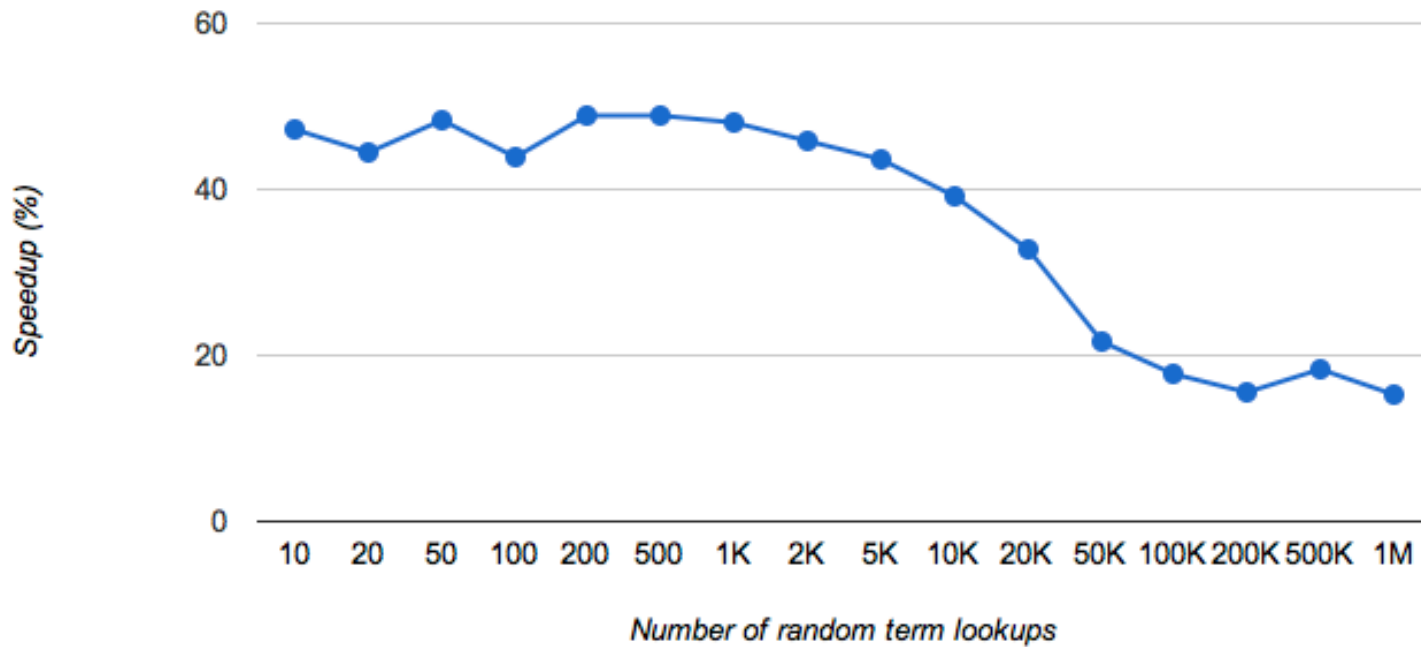
- Inlines low doc-freq terms into terms dict
- Saves extra seek to get the postings
- Excellent match for primary key fields, but also “normal” field (Zipf’s law)
- Wraps any other codec
- Likely default codec will use Pulsing
- <http://s.apache.org/JX3>



Progress not perfection.

Pulsing codec speedup

Pulsing Codec Speedup



SimpleText codec

- All postings stored in `_X.pst` text file
- Read / write
- Not performant
 - Do not use in production!
- Fully functional
 - Passes all Lucene/Solr unit tests (slowly...)
- Useful/fun for debugging
- <http://s.apache.org/eh>



Holding a grudge is like swallowing poison and waiting for the other person to die.

SimpleText codec

```
field body
  term bay
    doc 0
      pos 3
  term doors
    doc 0
      pos 4
  term hal
    doc 0
      pos 5
  term open
    doc 0
      pos 0
  term pod
    doc 0
      pos 2
  term the
    doc 0
      pos 1
```

END



Int block codec

- **Abstract codec**
 - Tests define Mock variable & fixed, with random block sizes
- **Encodes doc, frq, pos using block codecs**
 - Encoding/decoding block of ints at once
- **Fixed & variable blocks**
- **Easy to use: define flushBlock, readBlock**
- **Seek point requires pointer and block offset**



Fool me once, shame on you...

FOR/PFOR codec

- **Subclasses FixedIntBlock codec**
- **FOR (frame of reference) = packed ints**
 - eg: 1, 7, 3, 5, 2, 2, 5 needs only 3 bits per value
- **PFOR adds exceptions handling**
 - eg: 1, 7, 3, 5, 293, 2, 2, 5 encodes 293 as vInt
- **Not committed yet (LUCENE-1410)**
- **Initial results: ~20-40% speedup for many queries**
- **<http://s.apache.org/lw>**



Fool me twice, shame on me.

Other Codecs

- **PerFieldCodecWrapper**
- **AppendingCodec**
 - Never rewinds a file pointer during write
- **TeeSinkCodec**
 - Write postings to multiple destinations
- **FilteringCodec**
 - Filter postings as they are written
- **YourCodecGoesHereSoon**



Life is a series of one-way doors; pick yours carefully.

Agenda

- Who am I?
- Motivation
- New flex APIs
- Codecs
- **Wrap up**



Some ideas to try

- **In-memory postings**
 - Maybe only terms dict, select postings, etc.
- **Variable-gap terms index**
 - Add indexed term if $\text{docFreq} > N$
 - Good for noisy terms (eg, OCR)
- **DFA/trie/FST as terms dict/index**
- **Finer omitTFAP (OmitTF, OmitP, per-term)**
- **Block-encoding for terms dict sections**



The first investment is yourself.

Still to do

- Performance bottleneck of int block codecs
- Codec should include norms, stored fields, term vectors (LUCENE-2621)
- Enable serialization of attrs
- Switch to default hybrid (Pulsing, Standard, PForDelta) codec
- Expose codec configuration in Solr



Only the paranoid survive.

Summary

- **New 4D postings enum apis**
- **Pluggable codec lets you customize index format**
 - Many codecs already available
- **Goal 1 is realized: innovation is easy(ier)!**
 - Exciting time for Lucene...
- **Goal 2 is in progress...**
- **Sizable performance gains, RAM/GC reduction coming in 4.0**



¿Preguntas?



Backup



Composite vs atomic readers

- Lucene has aggressively moved to “per segment” search, starting at 2.9
- Flex furthers this!
- **Best to work directly with sub-readers**
 - Use direct flex APIs, eg `reader.fields()`, for this
- **If you must operate on composite reader...**
 - Use `MultiFields.getFields(reader)`, or
 - `SlowMultiReaderWrapper.wrap`
 - Beware performance hit!



Code: visit docs containing a term

```
Fields fields = reader.fields();
Terms terms = fields.terms("body");
TermsEnum iter = terms.iterator();
if (iter.seek(new BytesRef("pod")) ==
    SeekStatus.FOUND) {
    DocsEnum docs = iter.docs(null);
    int docID;
    while ((docID = docs.nextDoc()) !=
        DocsEnum.NO_MORE_DOCS) {
        ...
    }
}
```



Sep codec

- **Abstract**
- **Separates positions, freqs, docs, payloads, skip data into separate files**
- **Defines IntIndexInput/Output interface, used for positions, freqs, docs**
- **MockSep codec included in tests**

