

Apache Lucene 7

What's coming next?

Uwe Schindler

Apache Software Foundation | SD DataSolutions GmbH | PANGAEA

 @thetaph1 · uschindler@apache.org

My Background

- **Committer** and **PMC member** of **Apache Lucene and Solr** - main focus is on development of Lucene Core.
- Implemented fast numerical search and maintaining the new attribute-based text analysis API. Well known as *Generics and Sophisticated Backwards Compatibility* 🧐.
- **Elasticsearch** lover.
- Working as consultant and software architect at **SD DataSolutions GmbH** in Bremen, Germany.
- Maintaining **PANGAEA** (Data Publisher for Earth & Environmental Science) where I implemented the portal's geo-spatial retrieval functions with Apache Lucene Core and Elasticsearch.

ON THE WAY TO

Success

7...

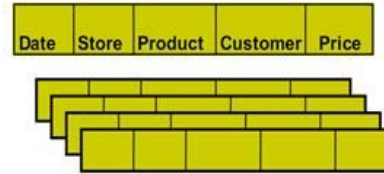


Lucene 7: When?

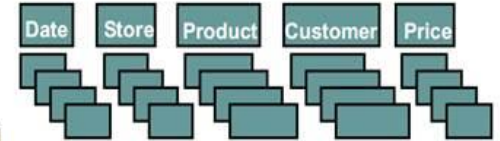
- Expected release date:
As always: no comment, but early summer is likely!
- **Release branch** (`branch_7x`) will be cut the next days

DOC VALUES

row-store

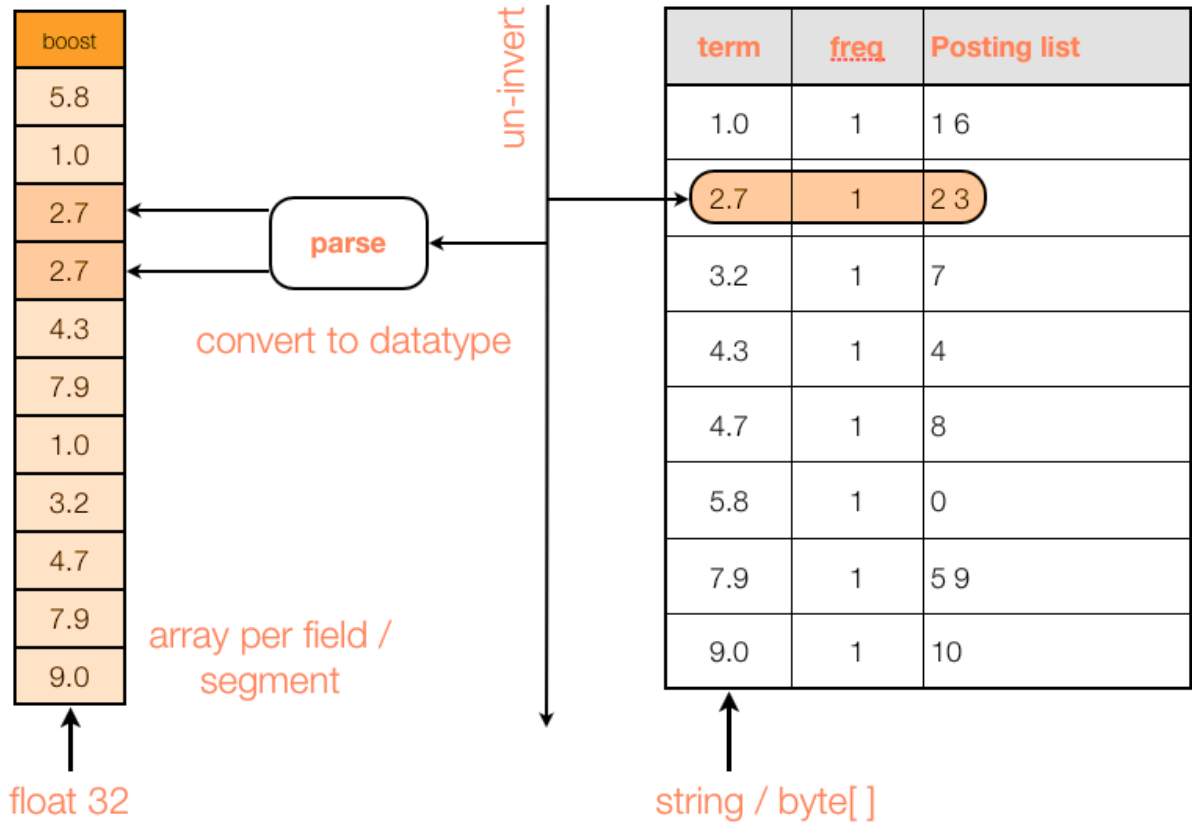


column-store



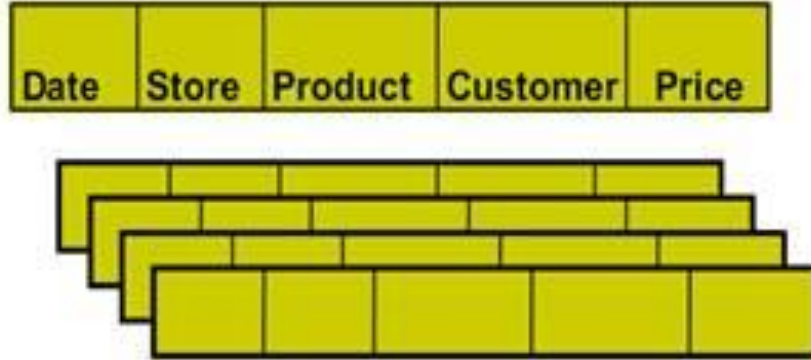
Lucene 4: DocValues

- FieldCache was already replaced by DocValues in **Lucene 4** !
- What are DocValues?

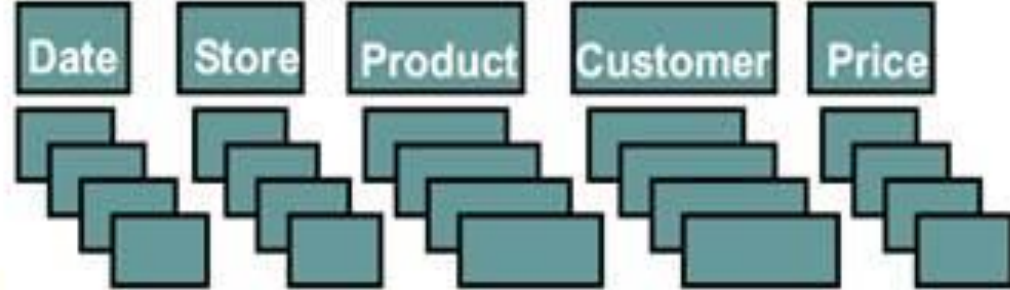


Lucene 4: DocValues

row-store



column-store




Lucene 7: DocValues changes

- Complete refactoring
- “random access” removed:

Solely Iterators !

Results: DocIdSetIterator



(A, 4)
(C, 12)
(H, 5)
(K, 19)
(L, 1)
(N, 2)

enqueue

TopDocs: PriorityQueue

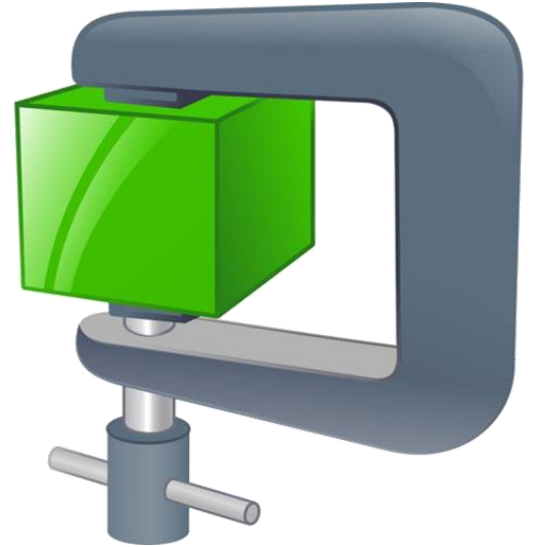
(K, 19)
(C, 12)
(H, 5)
(A, 4)

Bound!
(4 entries)

Lucene 7: DocValues changes

- Allows better index compression
- Document norms already optimized:

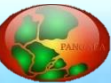
Sparse!



Lucene 7: DocValues changes

FieldCache
and
UninvertingReader
is finally gone!

NUMERICS



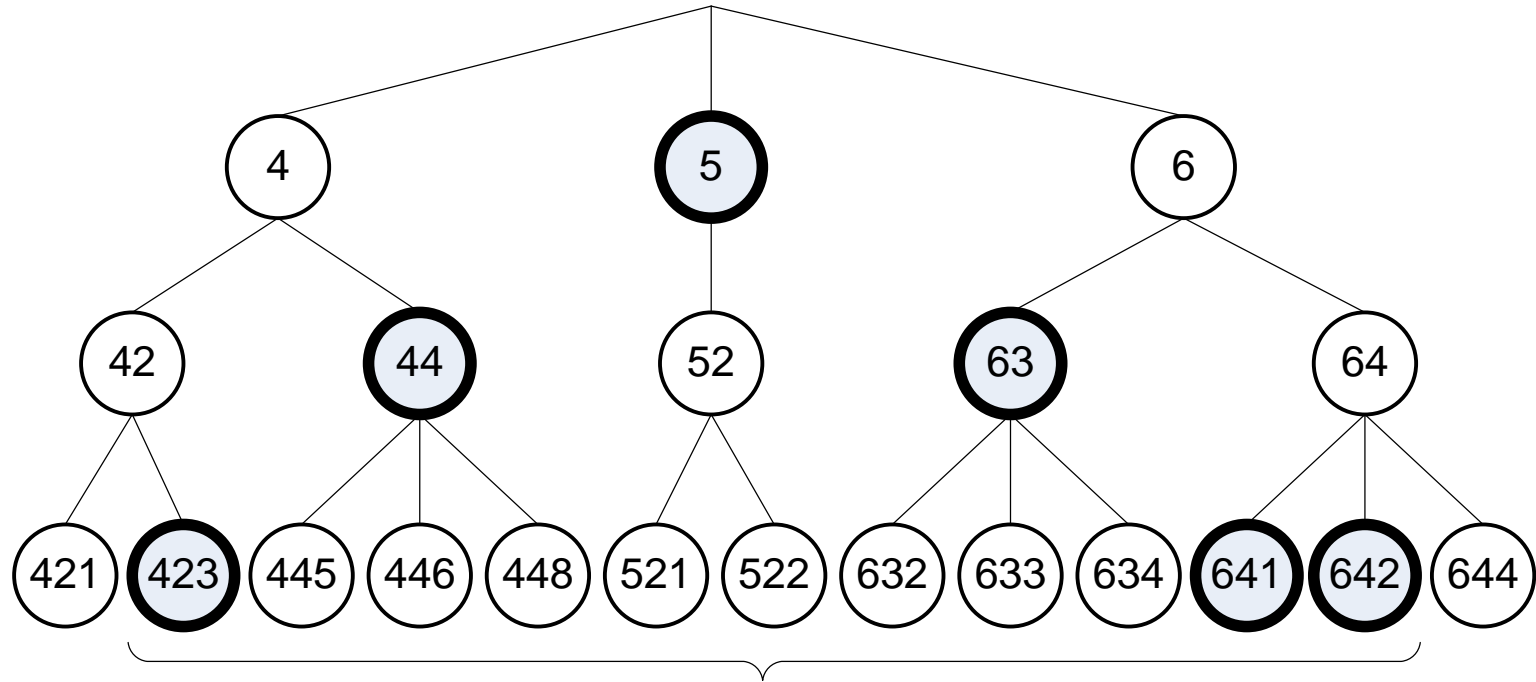
Lucene 6: Point Values

(also known as dimensional values)

- Successor of NumericField (Solr: TrieField)
- Multidimensional (e.g. geographic coordinates): 8 dims
- Up to 128 bits / 16 bytes per value (IPv6 range queries are now possible)

See: <https://www.elastic.co/blog/lucene-points-6.0>

Legacy Numeric Range Queries



Block k-d-Trees

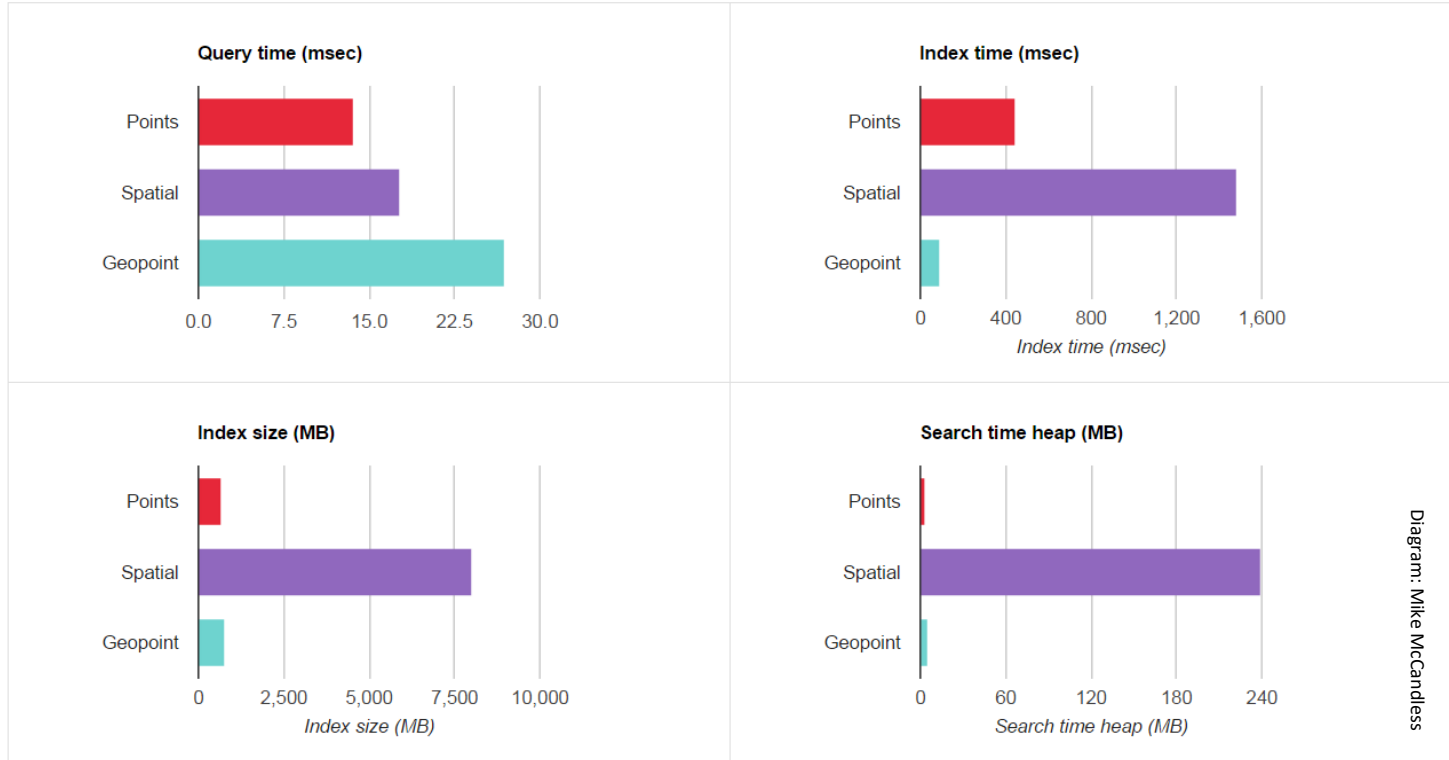
Very similar approach like **NumericField!**

- Just more dynamic
- Adapts dynamically depending on number of unique values!

Comparison (1D)



Comparison (2D)



Lucene 7: Legacy Numerics

- LegacyNumericField removed
- Terms in index no longer useable!

Lucene **7**: Legacy Numerics

- LegacyNumericField removed
- Terms in index no longer useable!

Requires reindexing!

SCORING

Lucene 6: Okapi BM25

It's now
the
default!



BTW: What's Okapi BM25 ???

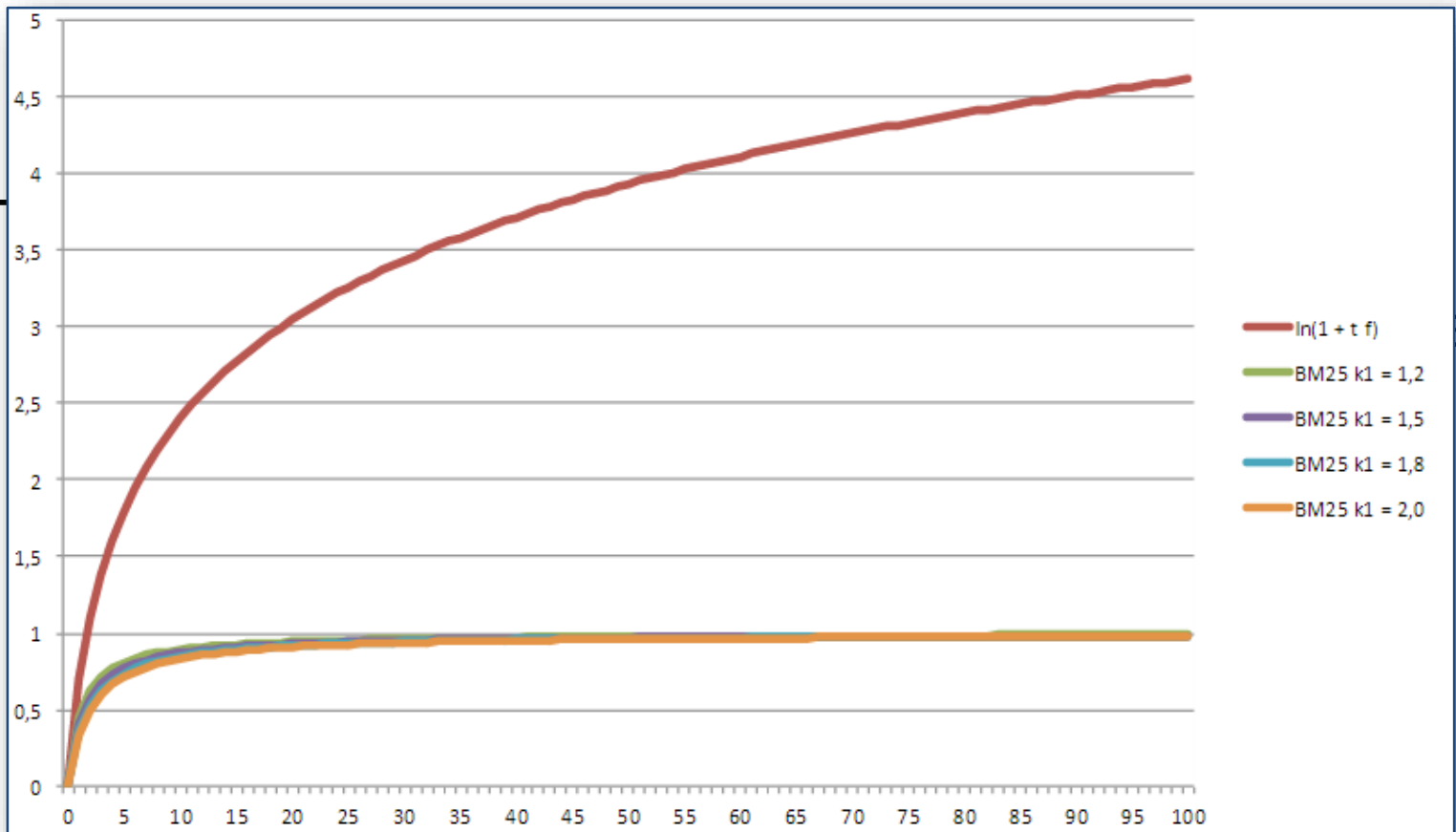
- BM25 is a bag-of-words retrieval function
- **probabilistic model** instead vector space model
- function of **TF** and **IDF**



BTW: What's Okapi BM25 ???

BTW: What's Okapi BM25 ???

- TF is not unbounded: saturation!
 - Documents with high term frequency don't increase score too much

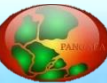


BTW: What's Okapi BM25 ???

- TF is not unbounded: saturation!
 - Documents with high term frequency don't increase score too much
- Average document length
 - Used with tuning factor to change significance of repeated terms

BTW: What's Okapi BM25 ???

- TF is not unbounded: saturation!
 - Documents with high term frequency don't increase score too much
- Average document length
 - Used with tuning factor to change significance of repeated terms
- IDF similar to standard TF-IDF



Lucene 7: BM25 followup

- Query normalization removed
- `BooleanQuery` coordination factors removed

Lucene 7: BM25 followup

- Query normalization removed
- `BooleanQuery` coordination factors removed
- Classical similarity aka “TF-IDF” no longer supported, but are still available!

Lucene 7: BM25 followup

- Query normalization removed
- `BooleanQuery` coordination factors removed
- Classical similarity aka “TF-IDF” no longer supported, but are still available!
- Index time boosting is gone

OTHER CHANGES

Lucene 7: Query Parser

- Default query parser no longer splits on whitespace
 - Splits only on operators
 - Better support for eastern languages
- Old behavior can be re-enabled

Lucene **7**: Custom term frequency

- New `TokenStream` attribute (integer):
TermFrequencyAttribute
- **1 by default**, allows to emulate “stacked tokens”
- Only works when **no positions/offsets** are indexed!

JAVA VERSION ?

Lucene 7: Java Version

- **Java 8** is still minimum requirement!
- `lucene-core.jar` only uses *compact1* profile
- All other (Lucene) parts use *compact2* profile

Lucene 7: Java 9 ?

- Compatibility with **Java 9** *module system* restrictions
- **Unicode 8:** 🍌 👮 🐸
 - with **ICU** or **Java 9**
 - **ICU** is now already on **Unicode 9!**
- **Nightly tests** with *early access builds*
 - currently **Java 9 build 173**



HOW TO MIGRATE ?

Lucene 7: "Anti-Feature"

Removal of Lucene 5 index support!



Lucene 7: "Anti-Feature"

Removal of Lucene 5 index support!

- Get rid of old index segments:
[IndexUpgrader](#) in latest Lucene 6 release helps!
- [Elasticsearch](#) has automatic index upgrader already implemented / [Solr](#) users have to manually do this



Lucene 7: Index Version

- Lucene stores version that created index
 - Preserved during merges or index upgrades
- Better detection of no longer supported features
 - Broken offset detection by default enabled for new indexes



THANK YOU!

Questions?



SD DataSolutions GmbH

Wätjenstr. 49

28213 Bremen, Germany

+49 421 40889785-0

<http://www.sd-datasolutions.de>

