LucenePapers

Lucene Papers

To understand the fundamental ideas behind Lucene, you should first get familiar with InformationRetrieval. This page tries to collect links to resources that explain some advanced topics.

Storage

Postings list encoding

In addition to VInt encoding, Lucene supports (or plans to support) other postings list encoding formats (FOR-delta, PFOR-delta, Simple9, ...).

- Performance of Compressed Inverted List Caching in Search Engines. Jiangong Zhang, Xiaohui Long, Torsten Suel. (2008)
- Lucene performance with the PForDelta codec. Mike McCandless, Changing bits, August 2nd, 2010.

The Pulsing codec

An optimized codec for fields that have lots of rare terms.

- Optimizations for Dynamic Inverted Index maintenance. Doug Cutting, Jan Pedersen.
- Lucene's PulsingCodec on "Primary Key" Fields. Mike McCandless, Changing bits, June 5th, 2010.

DocumentsWritersPerThread

Improved concurrency of index updates.

- Lucene indexing gains concurrency. Simon Willnauer, SearchWorkings blog (May 3rd, 2011),
- Exploiting full IO and CPU concurrency when indexing with Apache Lucene. Simon Willnauer, SearchWorkings blog (April 1st, 2011).

Query execution

Terms dictionary

In addition to its binary-search based terms dictionary, Lucene has a "block tree" terms dictionary, inspired of burst tries.

- LUCENE-3030 Block tree terms dict & index,
- Automata invasion Robert Muir, Michael McCandless,
- Burst Tries: A Fast, Efficient Data Structure for String Keys. Steffen Heinz , Justin Zobel , Hugh E. Williams. (2002)

NumericRangeQuery

Lucene has an optimized range query implementation for numeric types:

- NumericRangeQuery,
- Generic XML-based Framework for Metadata Portals. Computers & Geosciences 34 (12), 1947-1955. Schindler, U, Diepenbroek, M (2008).

BKD trees

BKD trees have been implemented to support geo capabilities in Lucene and have superseded NumericRangeQuery for one-dimensional data.

- Bkd-Tree: A Dynamic Scalable kd-Tree, Octavian ProcopiucPankaj K. AgarwalLars ArgeJeffrey Scott Vitter (2003)

Automaton-based fuzzy query

Lucene 4.0 supports an improved fuzzy query implementation that is based on Levenshtein automata.

- Fast String Correction with Levenshtein-Automata. Klaus Schulz , Stoyan Mihov. (2002)
- Lucene's FuzzyQuery is 100 times faster in 4.0. Mike McCandless, Changing bits, March 24th, 2011.

Scoring models

In addition to its default TF-IDF scoring algorithm, Lucene supports other scoring models such as Okapi BM25 and models based on language models.

- New index statistics in Lucene 4.0. Mike McCandless, Changing bits, March 14th, 2012,
- Okapi at TREC-3. Stephen E. Robertson, Steve Walker, Susan Jones, Micheline Hancock-Beaulieu, and Mike Gatford, In Proceedings of the Third Text REtrieval Conference (TREC 1994),
- Probabilistic models of information retrieval based on measuring the divergence from randomness, Gianni Amati and Cornelis Joost Van Rijsbergen (2002).

- The Probabilistic Relevance Framework: BM25 and Beyond, Stephen Robertson, Hugo Zaragoza (2009)

Incorporating non-textual signals into the final score

The below paper describes implementation ideas behind Lucene's FeatureField to fold non-textual static signals like pagerank, url length, etc. into the final score.

- Relevance Weighting for Query Independent Evidence, N. Craswell, S. Robertson, H. Zaragoza and M. Taylor (2005)

Block Max WAND

Block MAX WAND is an iteration over WAND that helps efficiently skip scoring non-relevant documents.

- Efficient query evaluation using a two-level retrieval process (WAND). Andrei Z. Broder, David Carmel, Michael Herscovici, Aya Soffer, and Jason Zien (2003)
- Faster Top-k Document Retrieval Using Block-Max Indexes. Shuai Ding, Torsten Suel (2011)
- From MaxScore to Block-Max WAND: The Story of How Lucene Significantly Improved Query Evaluation Performance. Adrien Grand, Jim Ferenczi, Robert Muir and Jimmy Lin (2020)

Misc

FST compression

Lucene uses FSTs a lot, so their in-memory size is important.

- Smaller Representation of Finite State Automata. Jan Daciuk, Dawid Weiss.

Twitter Earlybird

Modifications that Twitter made to Lucene to support lock-free updates and efficient early query termination for time-based relevance.

- Earlybird: Real-Time Search at Twitter, Michael Busch, Krishna Gade, Brian Larson, Patrick Lok, Samuel Luckenbill, and Jimmy Lin (2012).
- Earlybird - Realtime search @twitter. Talk by Michael Busch at Berlin Buzzwords (2012).