EFFICIENT TOP-K QUERY PROCESSING IN LUCENE 8

Tomoko Uchida

2019/02/26 @ Roppongi Hills

WHO AM I

- Twitter: @moco_beta
- 5+ years of experience w/ Solr and Elasticsearch
- Software Engineer @ AI Samurai Inc.
  - Developing patent search w/ AI technologies ☺
- Janome developer
- Luke: Lucene Toolbox Project co-mainteiner
- 改訂3版 Apache Solr 入門 lead author

Lucene/Solr 8.0 and Elasticsearch 7.0 coming...

SUMMARY OF THIS TALK

- Top-k query processing / scoring will be much faster!
- Especially effective in disjunction (OR) query
- Also works for complex queries such as PhraseQuery, WildcardQuery and their combinations
- Correct total hits count will not be returned (in default)

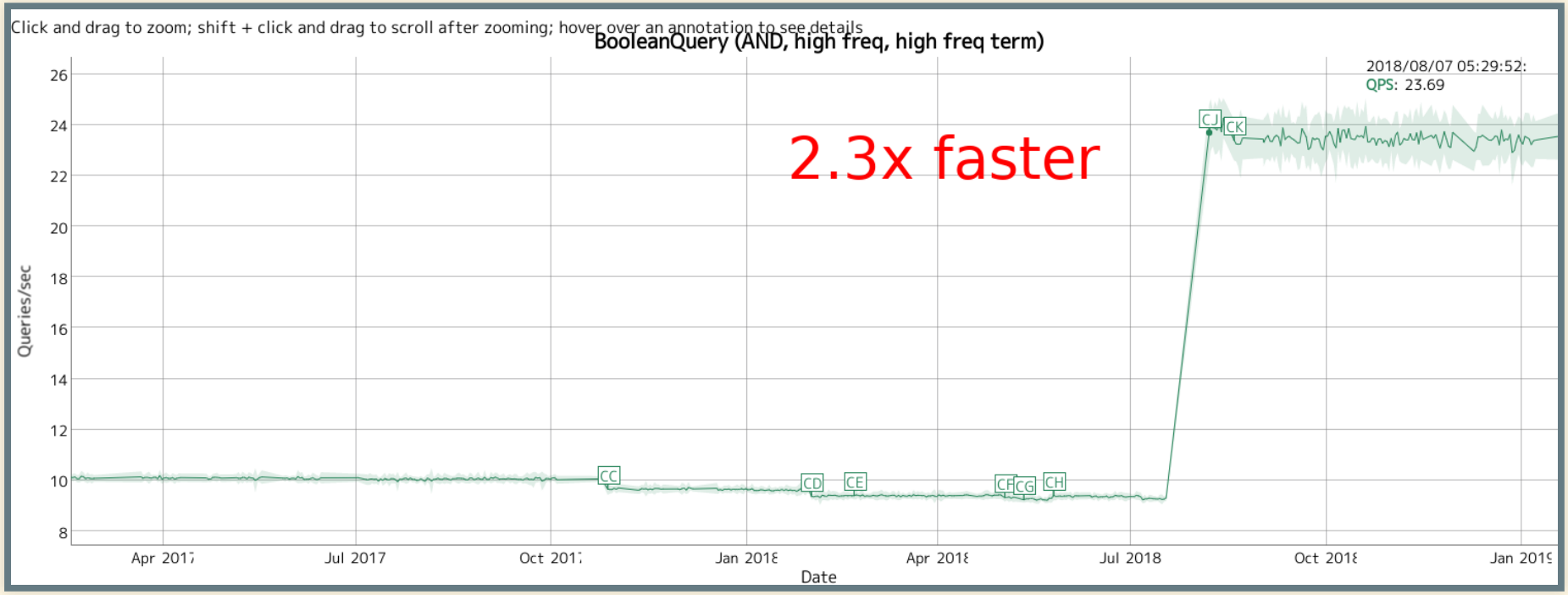
AND THERE IS A LONG VERSION...

This talk is a short version of my survey.

Please see this post (in Japanese) for more details :)

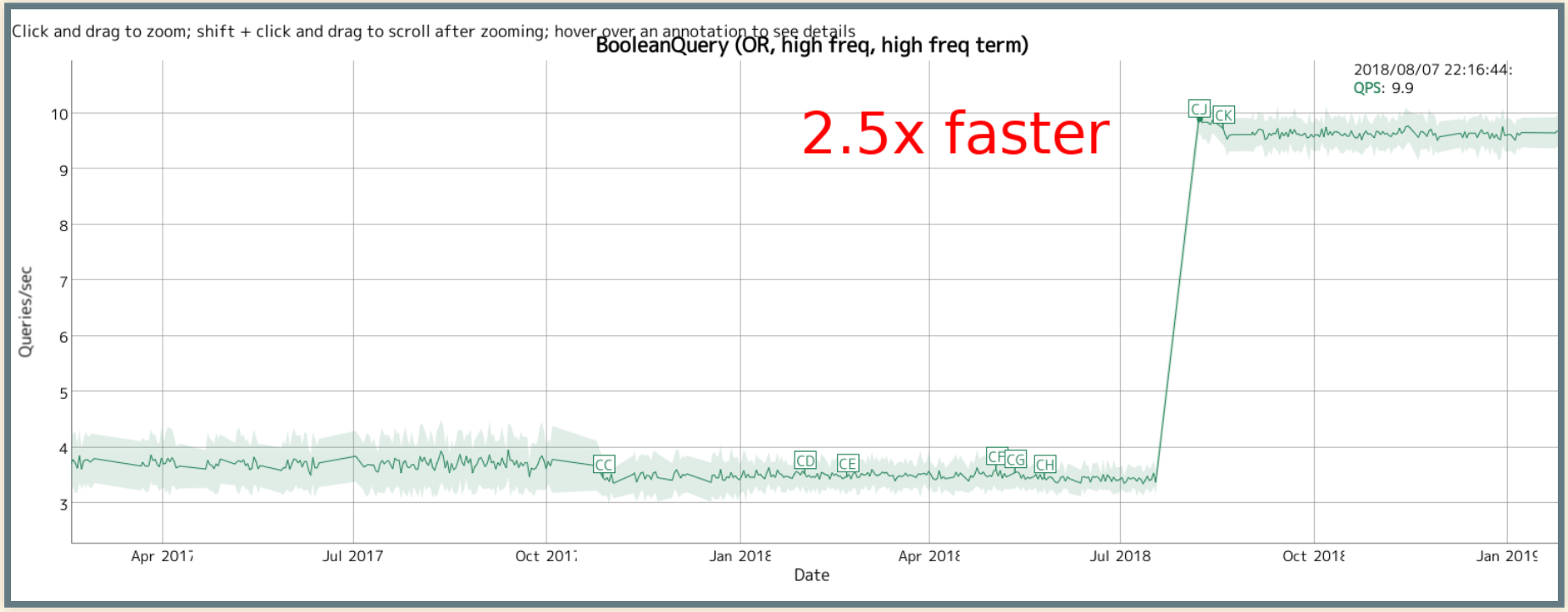
Lucene 8 の Top-k クエリプロセッシング最適化

HOW MUCH FASTER? - AND QUERY



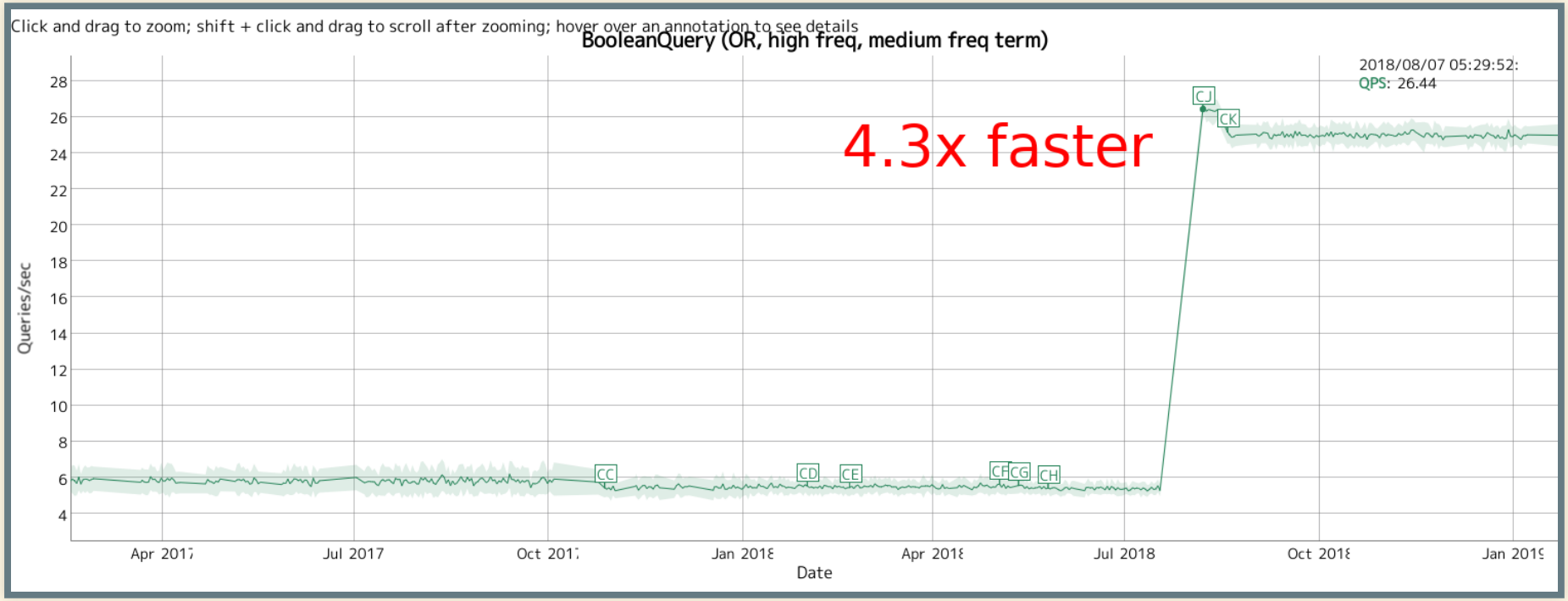
http://people.apache.org/~mikemccand/lucenebench/An

HOW MUCH FASTER? - OR QUERY (1)



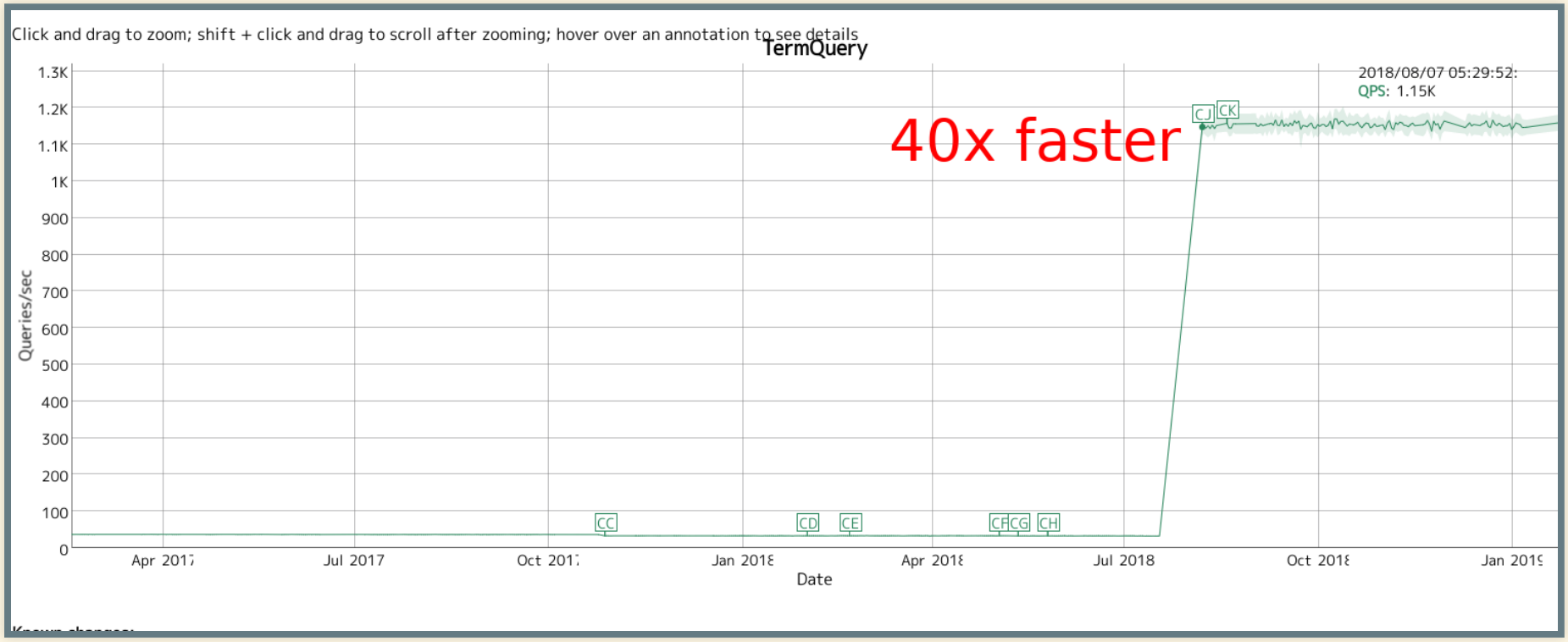
http://people.apache.org/~mikemccand/lucenebench/Or

HOW MUCH FASTER? - OR QUERY (2)



http://people.apache.org/~mikemccand/lucenebench/Or

HOW MUCH FASTER? - TERM QUERY



http://people.apache.org/~mikemccand/lucenebench/Te

REFERENCES

- Magic WAND: Faster Retrieval of Top Hits in Elasticsearch
- (FOSDEM 2019) Super-speedy scoring in Lucene 8
- (FOSDEM 2019) Apache Lucene and Apache Solr 8
- (Berlin Buzzwords 2012) Efficient Scoring in Lucene
- 転置インデックスと Top k-query

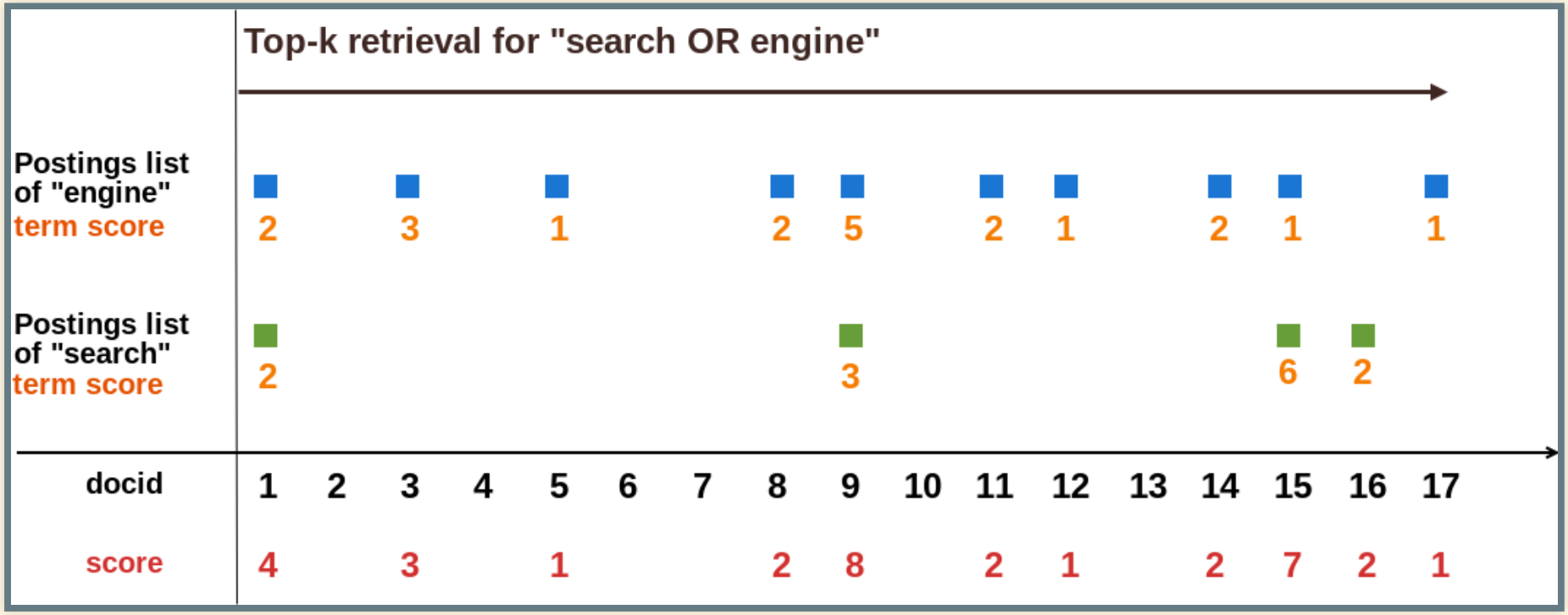
PAPERS

- [1] T. Strohman, H. Turtle, and B. Croft. Optimization strategies for complex queries. In Proceedings of ACM SIGIR conference, 2005.
- [2] K. Chakrabarti, S. Chaudhuri, V. Ganti. Interval-Based Pruning for Top-k Processing over Compressed Lists, in Proc. of ICDE, 2011.
- [3] A. Z. Broder, D. Carmel, M. Herscovici, A. Soffer, J. Y. Zien. Efficient Query Evaluation using a Two-Level Retrieval Process, in Proc. of CIKM, 2003.
- [4] S. Ding and T. Suel. Faster top-k document retrieval using block-max indexes. SIGIR, 2011.

ALGORITHMS

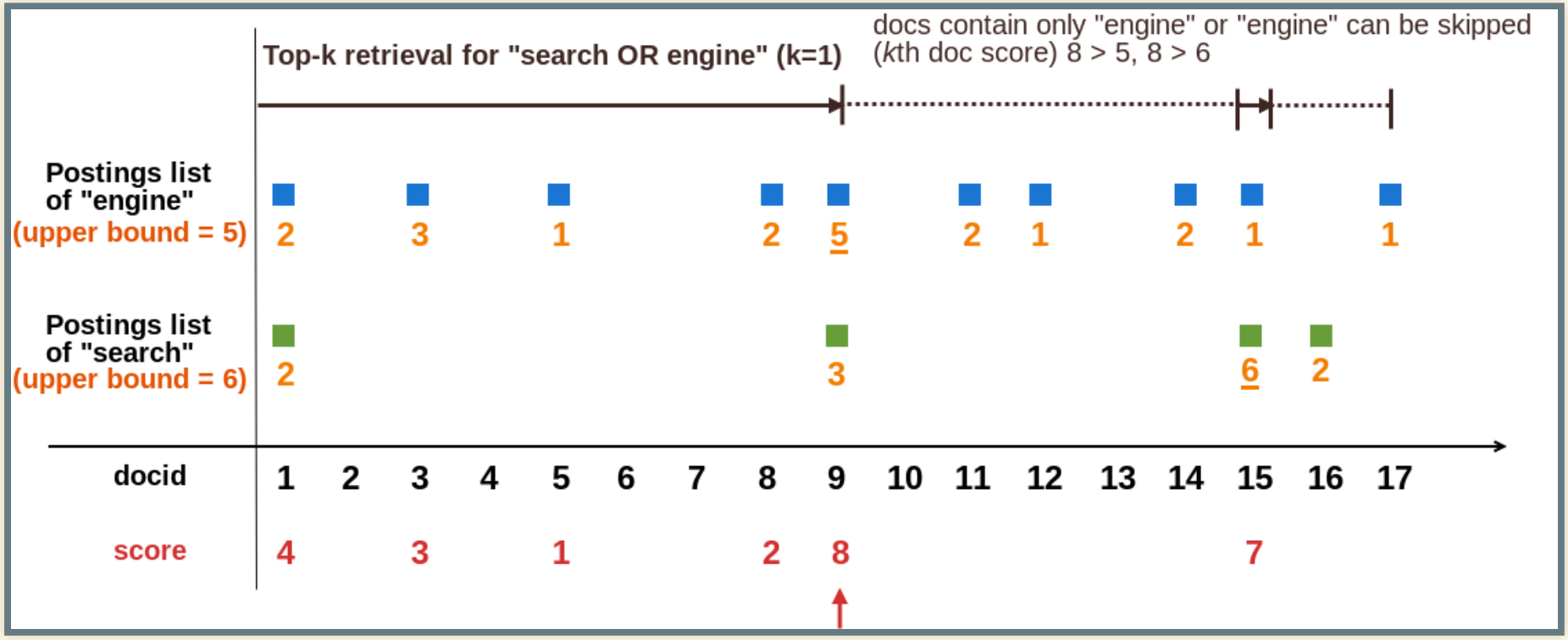
POSTING LIST RETRIEVAL AND THE CHALLENGE ON DISJUNCTION

Query "search OR engine"



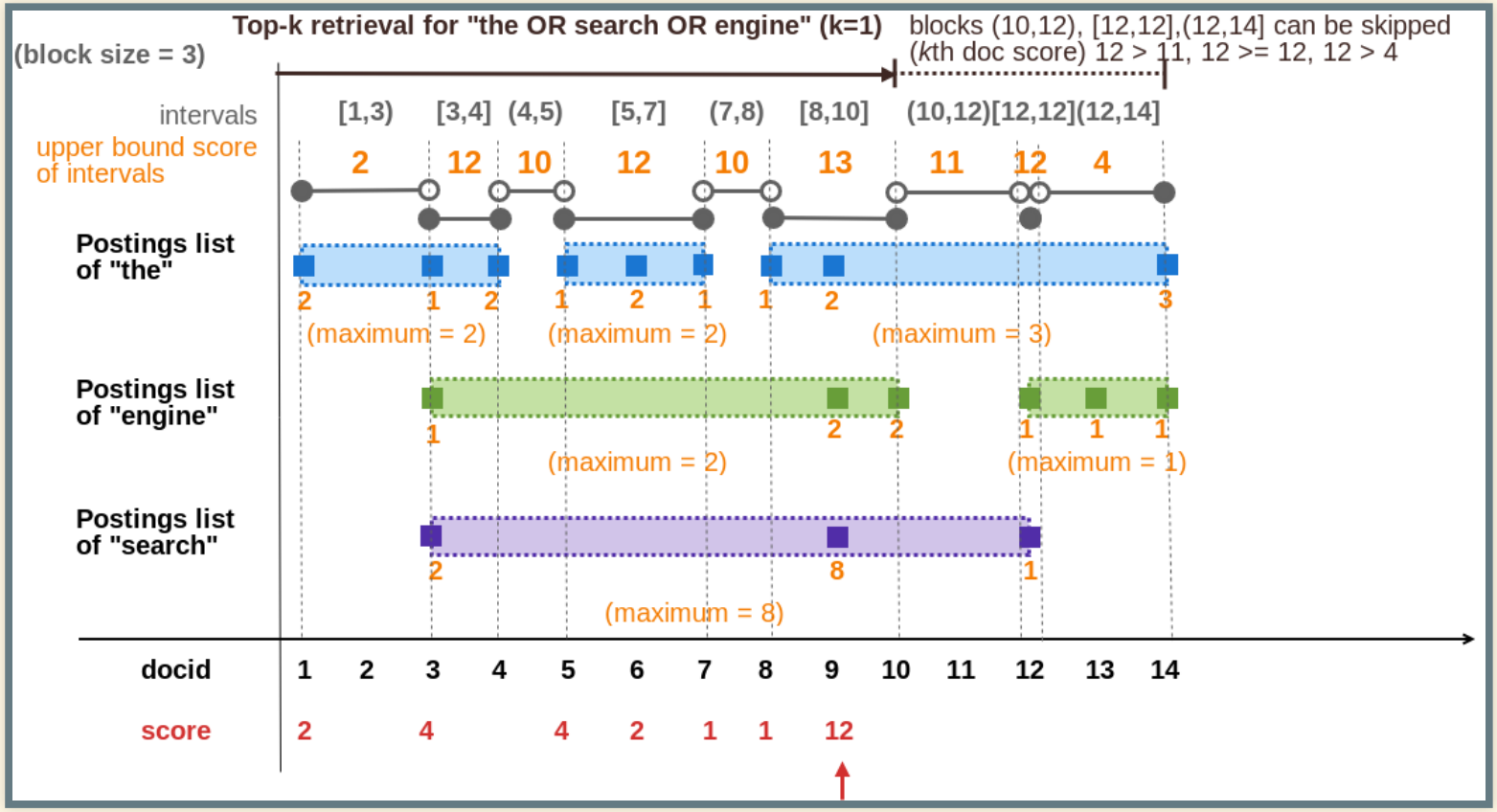
MAXSCORE

- Introduced by H.R.Turtle and J.Flood in 1995



INTERVAL-BASED PRUNING

- MaxScore variant adopted to block compressed indexes [2]



WAND

- Special operator proposed in [3]
- "WAND" is the abbreviation for "Week AND" or "Weighted AND"
- OR is being close to AND when a document contains a large enough subset of the query terms
- Score of a document having a large subset of the query terms is higher than the ones of documents with a few of them

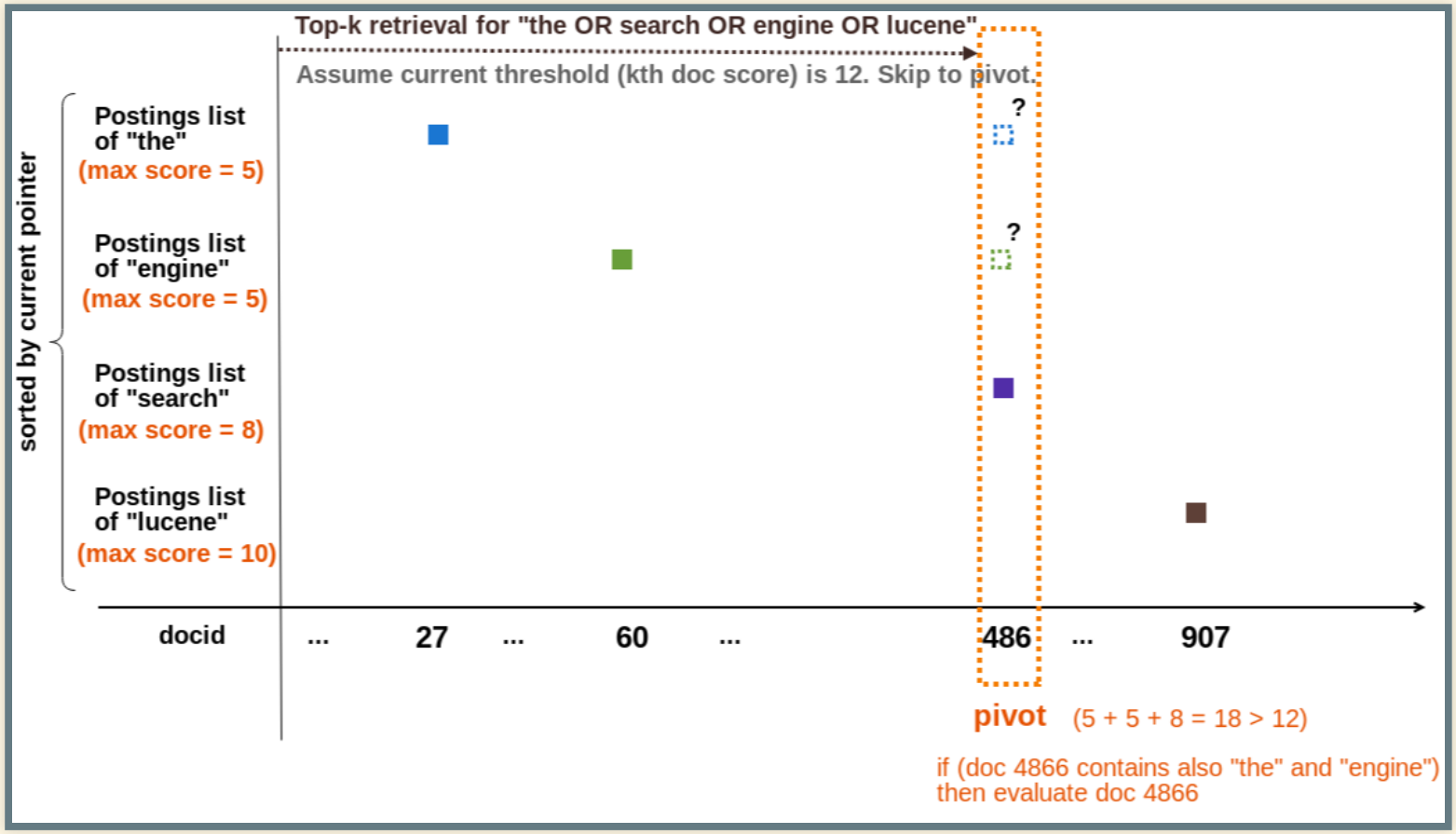
SOUNDS FAMILIAR?

Lucene already has similar concept :

"Minimum Should Match"

WAND

Query "the OR search OR engine OR lucene"



WAND

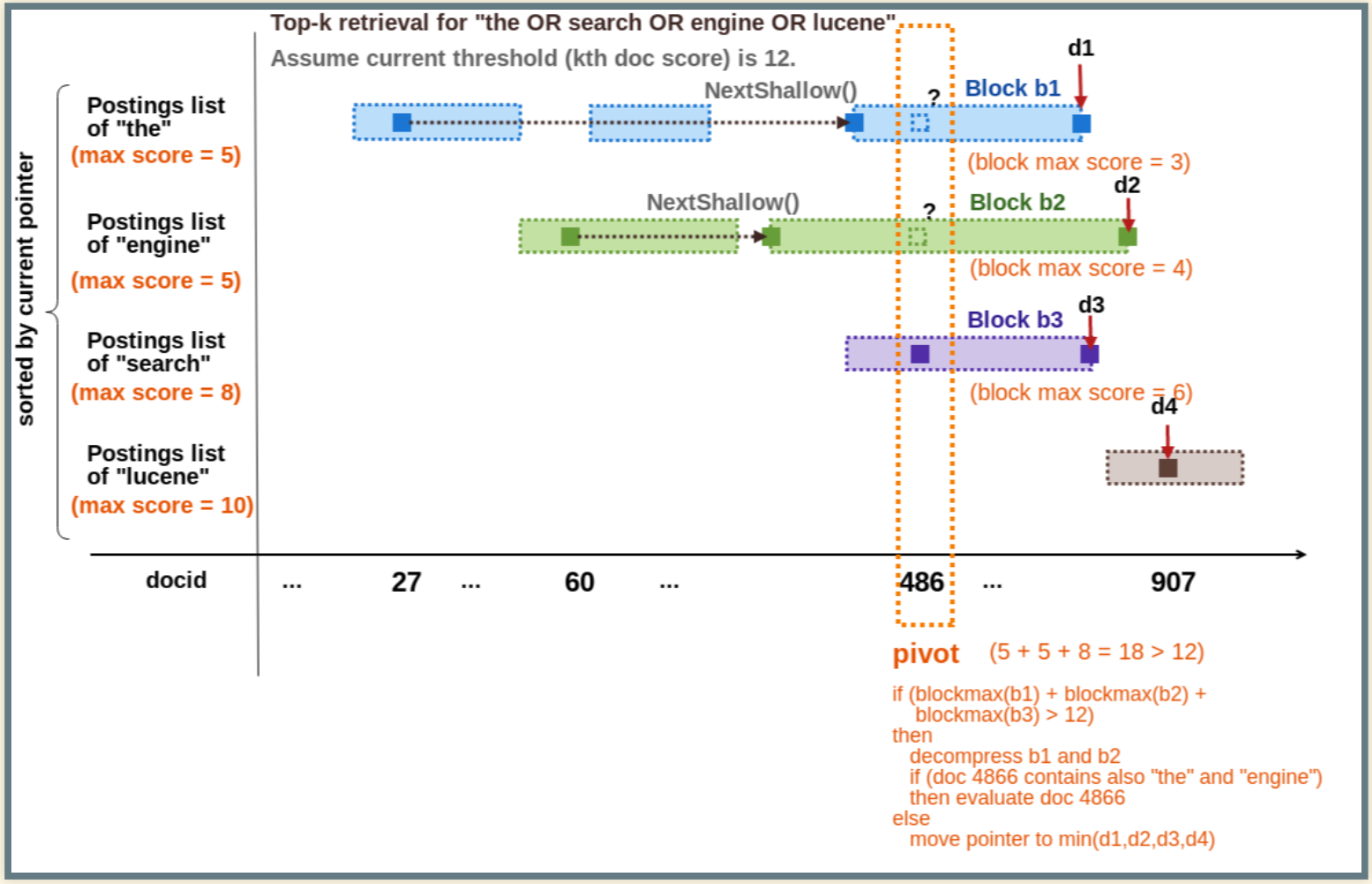
Steps

1. Assume current threshold (kth highest score) is 12.
2. Sort postings by current pointer.
3. Find "pivot" term and docid - here, that is "search" and id=486.
4. Calculate the partial score for doc 486 if it also contains "the" and "engine".

BLOCK-MAX WAND

- WAND variant working with block compressed indexes [4]
- Finally come in Lucene!

BLOCK-MAX WAND



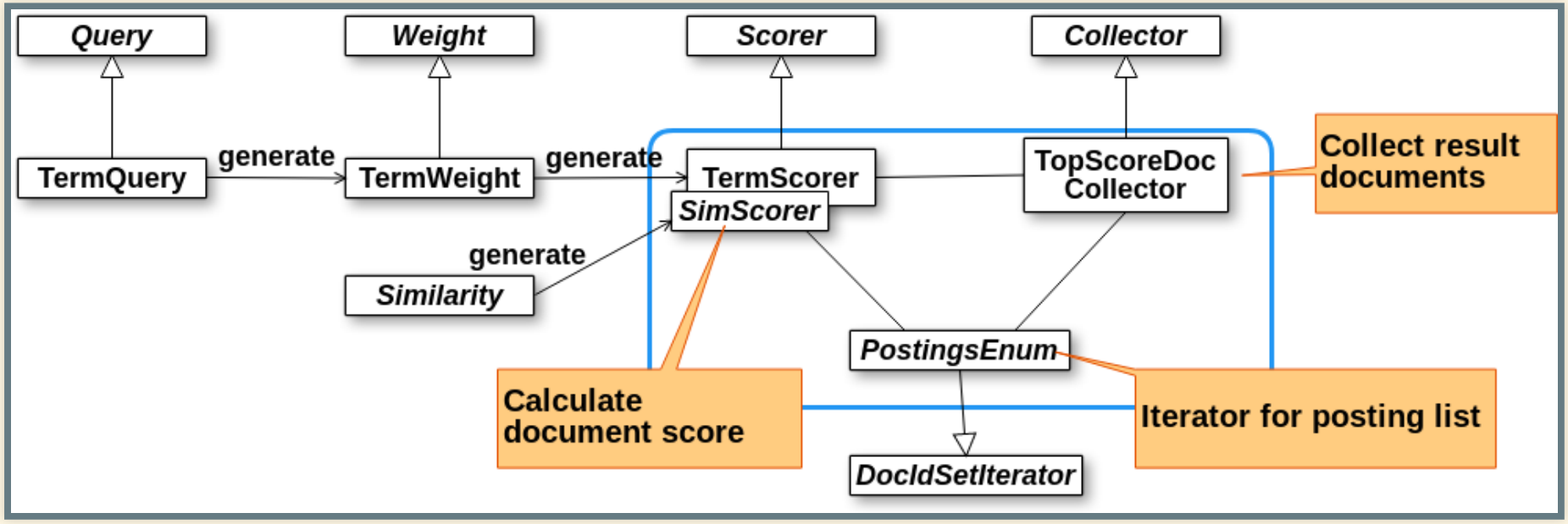
DIVE INTO IMPLEMENTATION

DISCLAIMER

- This is about low-level, complex part of Lucene. Could include mistakes... ☺
- Lucene API can be rapidly changed. This is based on branch_8_0 branch.

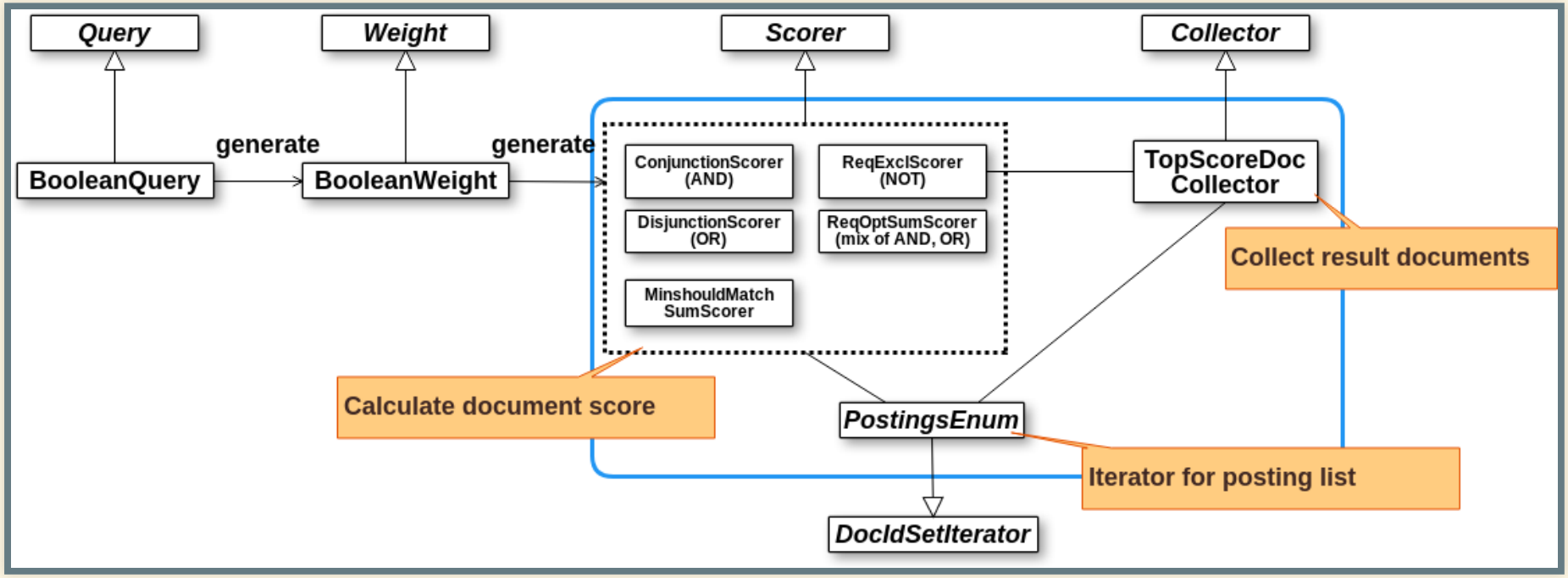
REVIEW: LUCENE SCORING ARCHITECTURE

Ex. TermQuery



REVIEW: LUCENE SCORING ARCHITECTURE

Ex. BooleanQuery



BLOCK-MAX WAND IMPLEMENTATION

Changes in indexing

- o.a.l.index.Impact
- o.a.l.codecs.CompetitiveImpactAccumulator
- o.a.l.codecs.lucene50.Lucene50SkipWriter#writeImpa
- ...

BLOCK-MAX WAND IMPLEMENTATION

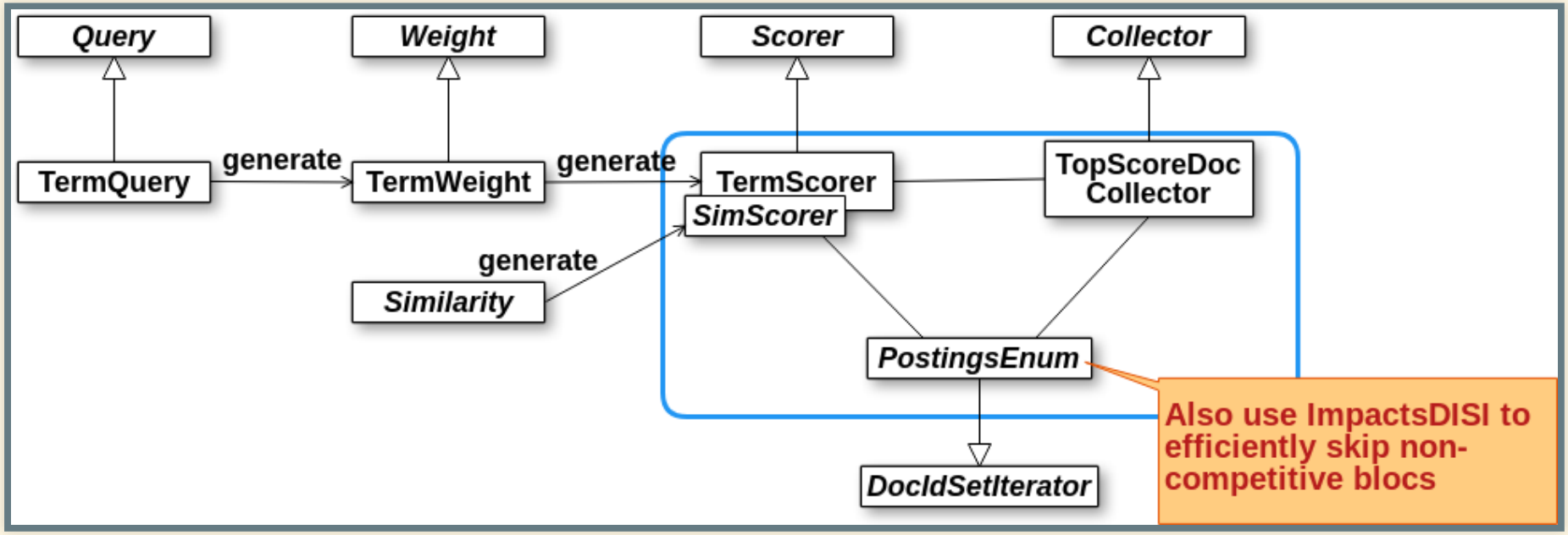
Changes in retrieving posting list

- o.a.l.codecs.lucene50.Lucene50ScoreSkipReader
- o.a.l.index.ImpactsSource
- o.a.l.search.MaxScoreCache
- o.a.l.search.ImpactsDISI
- ...

BLOCK-MAX WAND IMPLEMENTATION

Changes in scoring

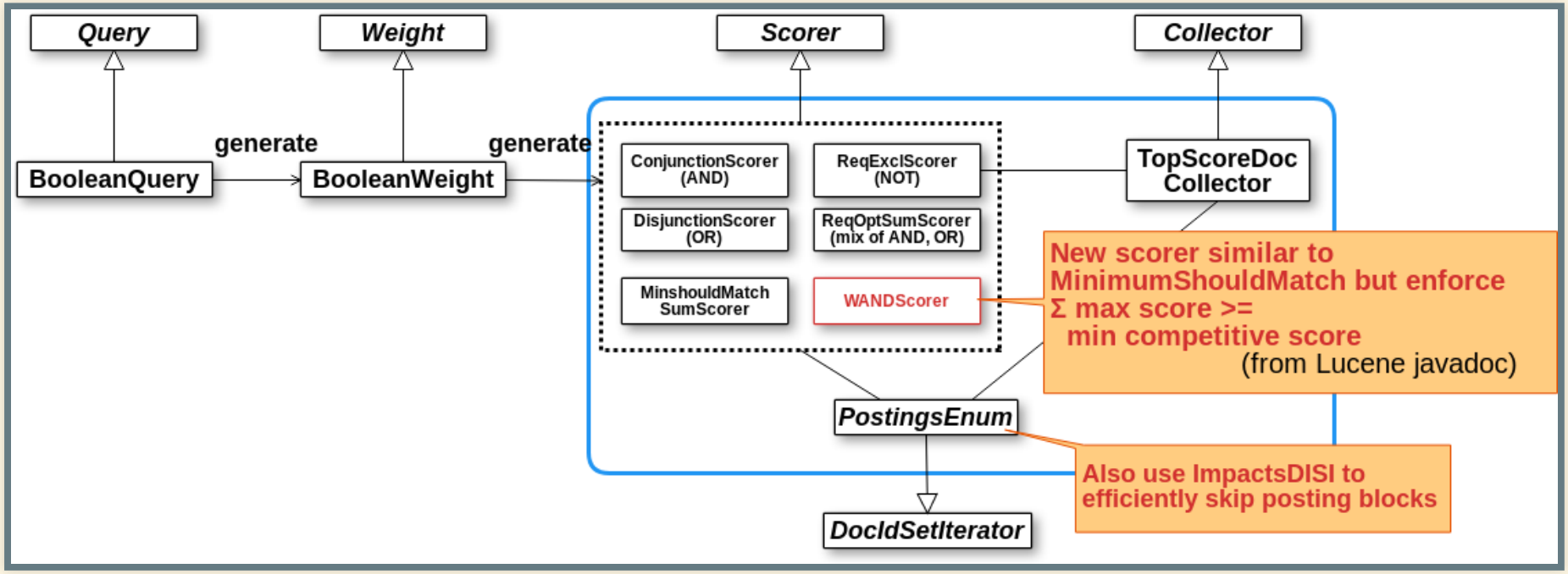
Ex. TermQuery



BLOCK-MAX WAND IMPLEMENTATION

Changes in scoring

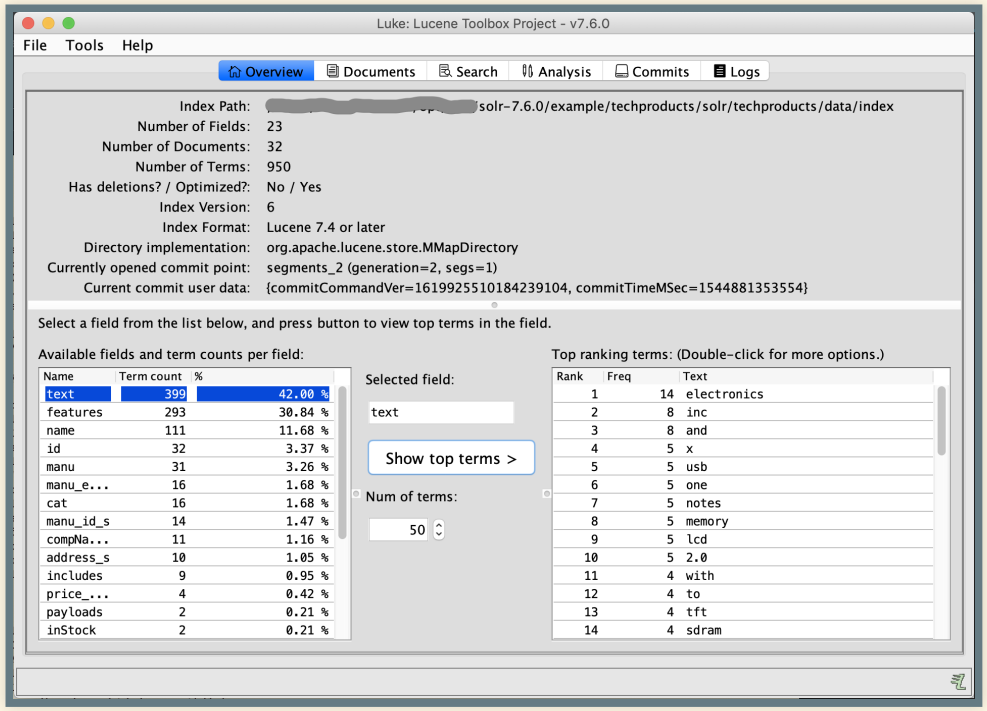
Ex. BooleanQuery



[ANN] LUKE HAS BEEN REVISED!

GUI tool for introspecting and debugging your Lucene/Solr/Elasticsearch index.

https://github.com/DmitryKey/luke



[ANN] LUKE HAS BEEN REVISED!

- Eventually rewritten on top of Swing ... in 2019? It's a long story :)
- Licenced under ALv2 and works fine with JDK11+
- Popular in US, Europe and China
- Still big growth potential in Japan

DmitryKey / **luke**
forked from sonarme/luke

Unwatch 140 | Unstar 1,122 | Fork 301

THANK YOU ☺

Happy (paper | code) reading!