elastic{on} 2017

Get the Lay of the (Lucene) Land

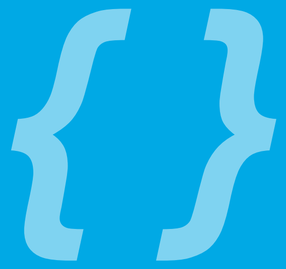
Adrien Grand

Elastic

March 8th 2017

@jpountz

Working with Lucene since 2010
Lucene committer since 2012
Lucene PMC since 2013
Elastic employee since 2013



Apache Lucene is a free and open-source information retrieval software library

Wikipedia

Where is Lucene heading?

Lucene 4 (2012)	Doc values - Flexible scoring Better postings/store compression Fuzzy queries speedup
Lucene 5 (2015)	Index safety Slow query execution
Lucene 6 (2016)	Points - Index sorting BM25 by default Multi-term synonyms
Lucene 7 (2017?)	Query planning Sparse doc values

Where is Lucene heading? Analytics

Lucene 4 (2012)	Doc values - Flexible scoring Better postings/store compression Fuzzy queries speedup
Lucene 5 (2015)	Index safety Slow query execution
Lucene 6 (2016)	Points - Index sorting BM25 by default Multi-term synonyms
Lucene 7 (2017?)	Query planning Sparse doc values

Where is Lucene heading? Structured search

Lucene 4 (2012)	Doc values - Flexible scoring Better postings/store compression Fuzzy queries speedup
Lucene 5 (2015)	Index safety Slow query execution
Lucene 6 (2016)	Points - Index sorting BM25 by default Multi-term synonyms
Lucene 7 (2017?)	Query planning Sparse doc values

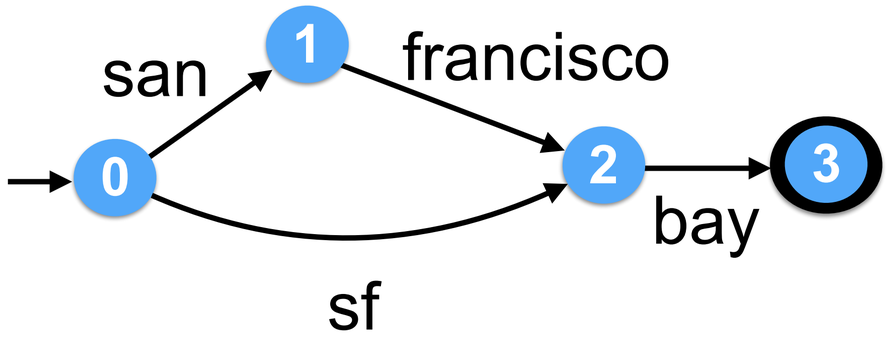
Where is Lucene heading? Data store

Lucene 4 (2012)	Doc values - Flexible scoring Better postings/store compression Fuzzy queries speedup
Lucene 5 (2015)	Index safety Slow query execution
Lucene 6 (2016)	Points - Index sorting BM25 by default Multi-term synonyms
Lucene 7 (2017?)	Query planning Sparse doc values

Better query parsing (6.2-7.0+)

- Query parsers no longer split on whitespace
  - up to the search analyzer
- Correct multi-term synonyms at query time

SF bay

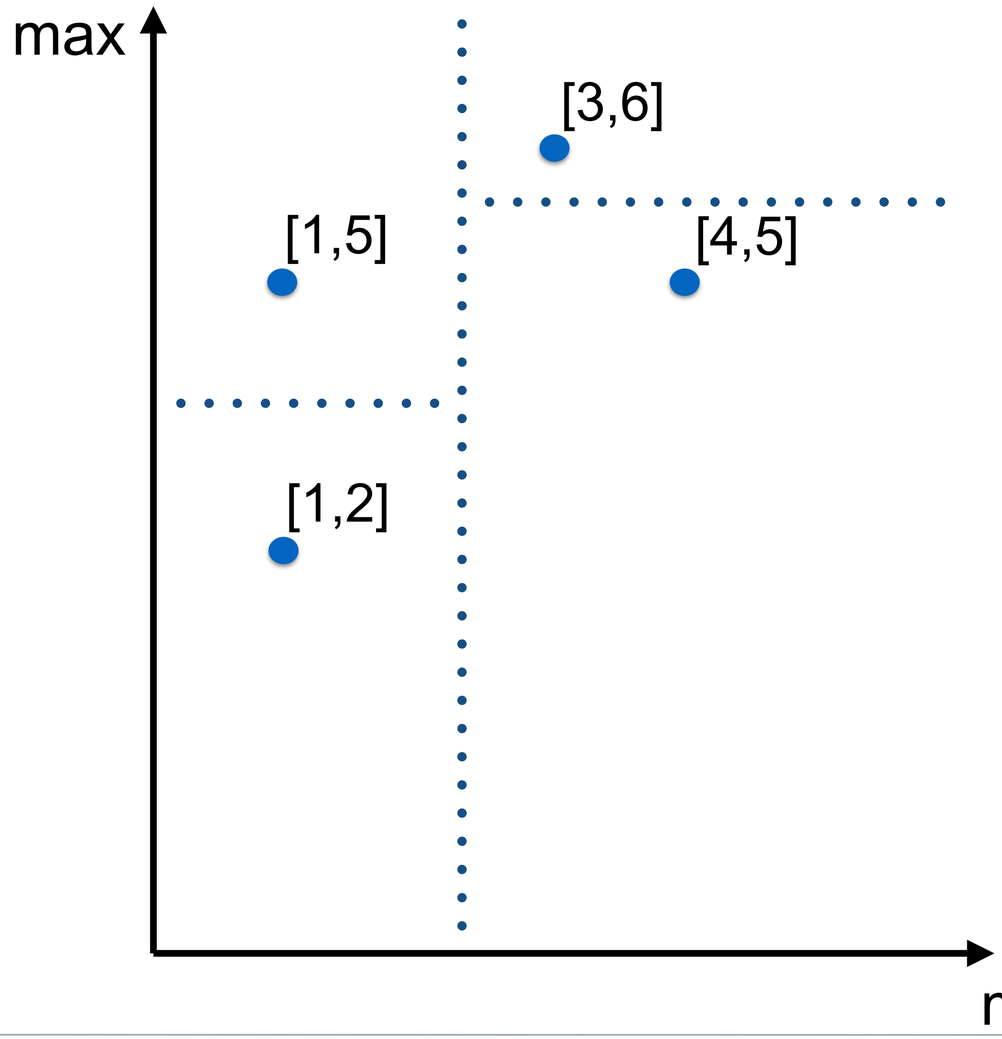


(sf OR “san francisco”)
OR bay

More information on Thursday at 12:45
Elasticsearch search improvements
by Jim Ferenczi

Range fields (6.2+)

- Indexed like 2D points in a BKD tree
- More efficient than 2 separate 1D ranges
- INTERSECTS / WITHIN / CONTAINS / CROSSES relations



More information on Thursday at 12:45
Elasticsearch search improvements
by Nick “geo” Knize

Index sorting (6.2+)

- Queries return documents in index order
- Index sorting makes index order configurable
- Benchmark on the geonames dataset
  - 8.5 M documents

```
{
    “geoname_id”: 6252001,
    “name”: “United States”,
    “type”: “country”,
    “country_code”: “US”,
    “population”: 310232863
}
```

Index sorting: faster sorting (6.2+)

INDEX ORDER	RANDOM ORDER	POPULATION DESC
INDEX TIME	64s	87s (+36%)
INDEX SIZE	463MB	436MB (-6%)
TOP 10 LOCATIONS BY POPULATION	120ms	0.02ms (6000x faster)
IDEM + HIT COUNT	120 ms	17ms (7x faster)

Index sorting: faster searching (6.2+)

INDEX ORDER	RANDOM ORDER	TYPE ASC, COUNTRY_CODE ASC
INDEX TIME	64s	136s (+112%)
INDEX SIZE	463MB	374MB (-19%)
TYPE:(CITY OR COUNTRY)	40ms	13ms (3x faster)
TYPE:CITY AND COUNTRY_CODE:US	46ms	28ms (1.6x faster)

Sparse doc values fields (7.0+)

Doc ID	Value
0	42
1	
2	
3	-3
4	100
5	

6.x storage

Docs with field	[T, F, F, T, T, F]
Value	[42, 0, 0, -3, 100, 0]

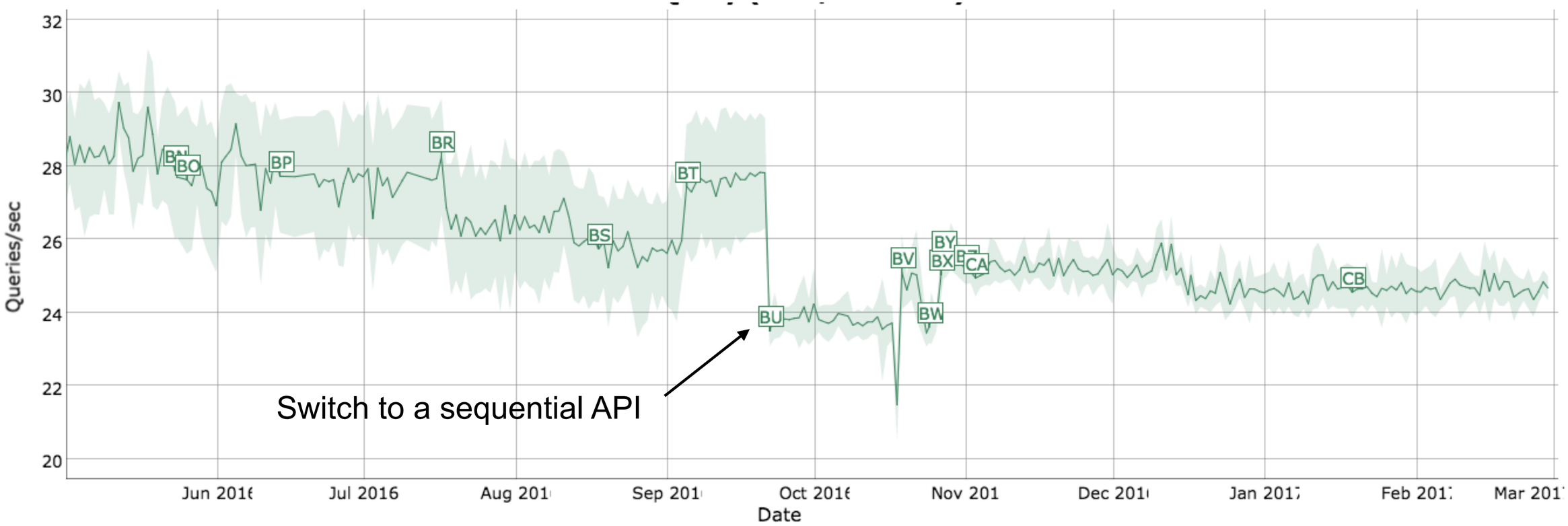
7.0 storage

Docs with field	[0, 3, 4]
Value	[42, -3, 100]

Sparse doc value fields (7.0+)

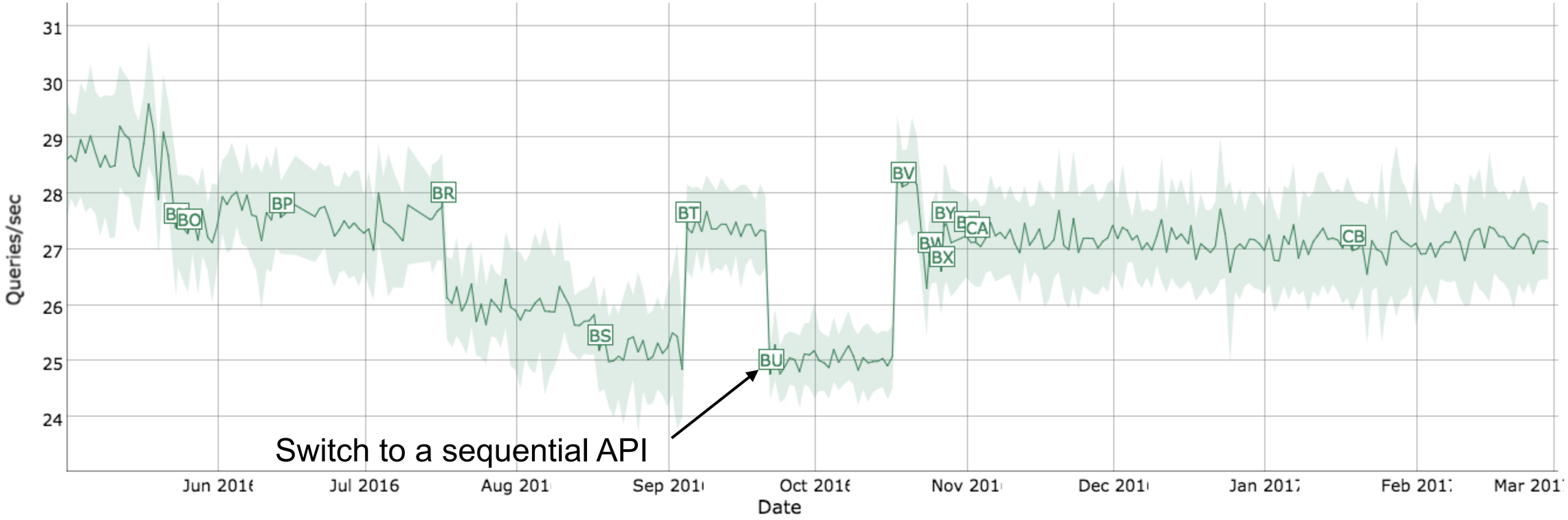
- **Pros**
  - More space-efficient
  - Faster merging
  - More potential for compression
- **Cons**
  - Only sequential access is efficient
  - 0-10% slow down for sorting

TermQuery (date/time sort)



http://people.apache.org/~mikemccand/lucenebench/TermDTSort.html

TermQuery (title sort)



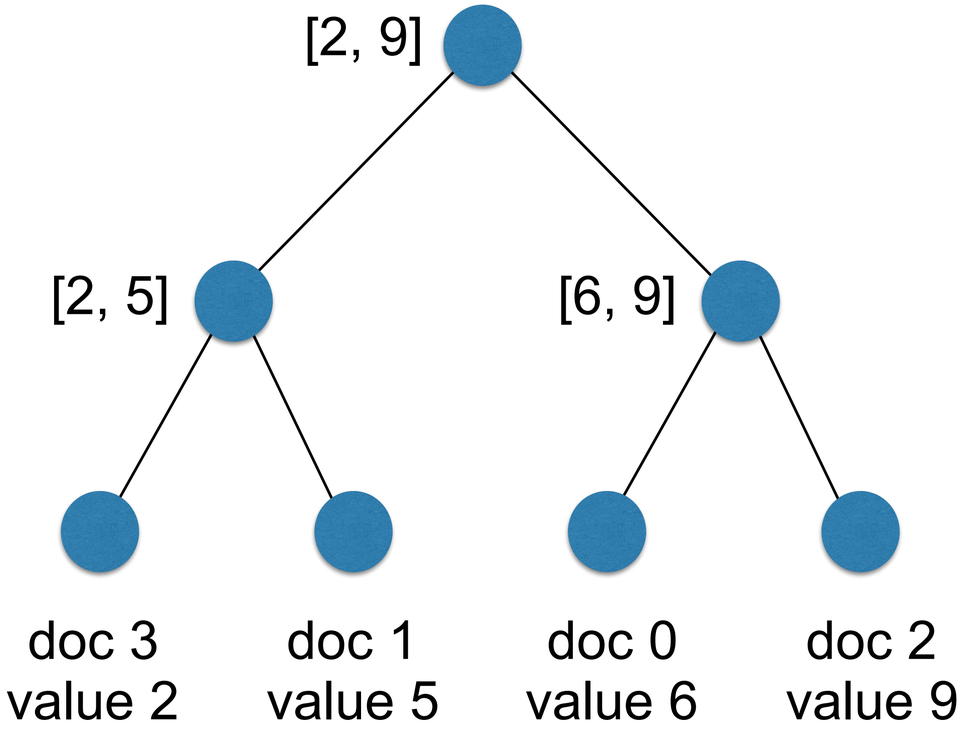
http://people.apache.org/~mikemccand/lucenebench/TermTitleSort.html

Query planning (6.5+)

- Queries have 2 primitive operations:
  - find matches
  - verify matches
- Conjunction (ANDed clauses):
  - 1 clause that finds matches
  - 1-N clauses that verify matches

Range query: points

- Find all matches?
  - O(#matches)
- Verify N matches?
  - O(N + #matches)



Range query: doc values

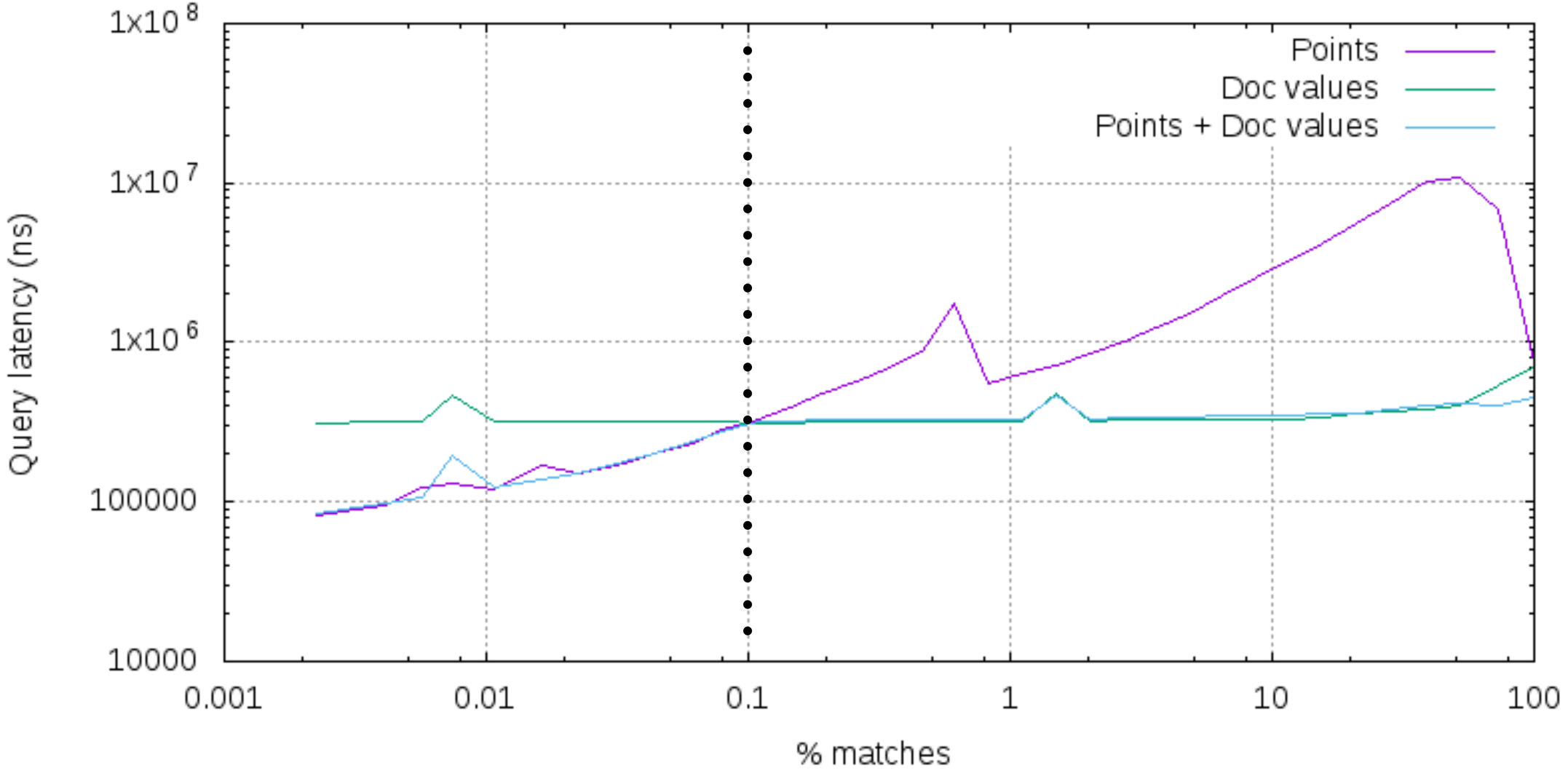
- Find all matches?
  - O(#docs) (linear scan)
- Verify N matches?
  - O(N)

Doc ID	Value
0	6
1	5
2	9
3	2

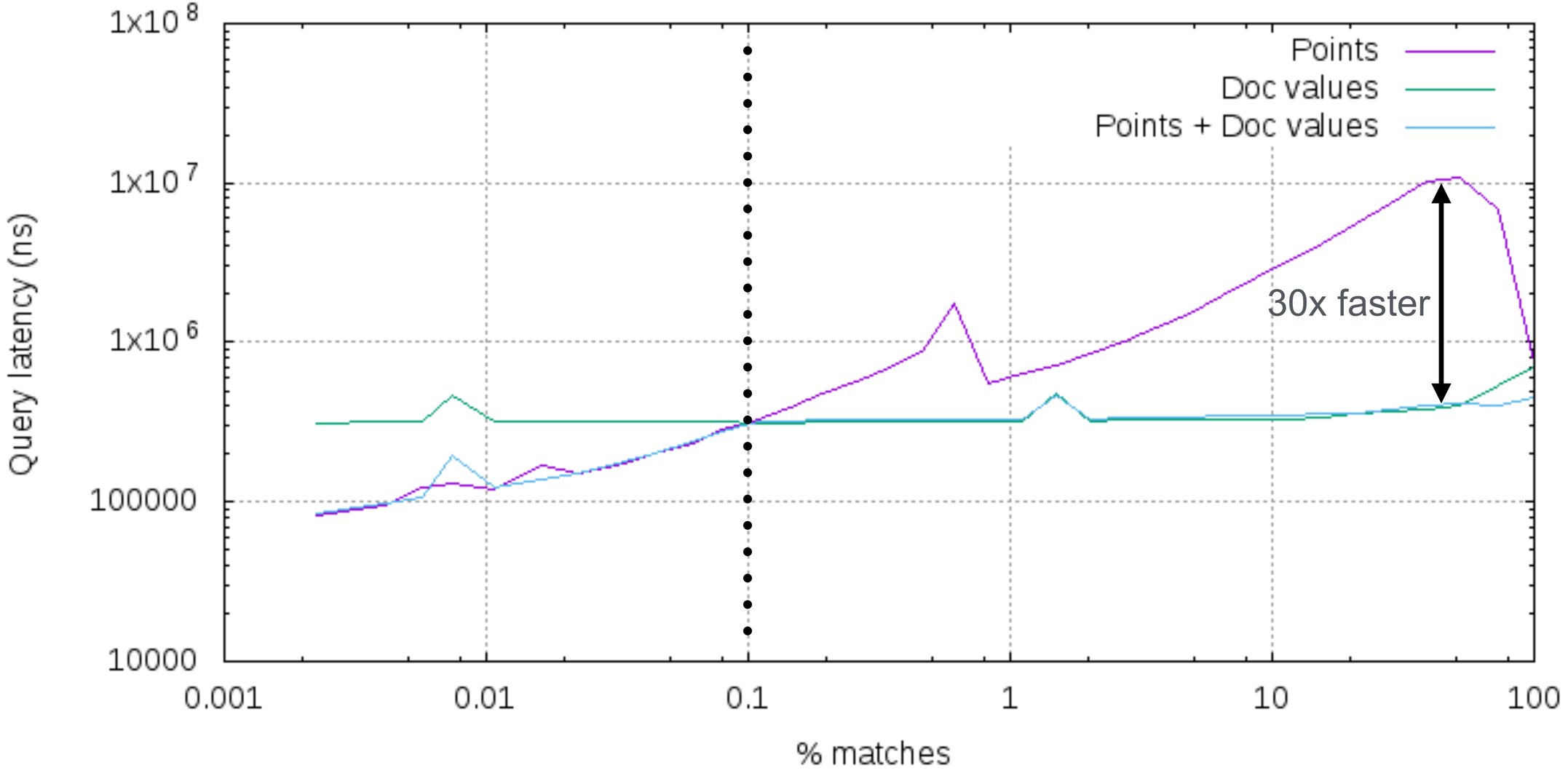
Query planning: benchmark

- 10M wikipedia subset
  - body: text
  - last edit: date
- Query: full-text query on body, filtered by a date range on the last edit date
- Query planning:
  - points if range is more selective
  - doc values otherwise

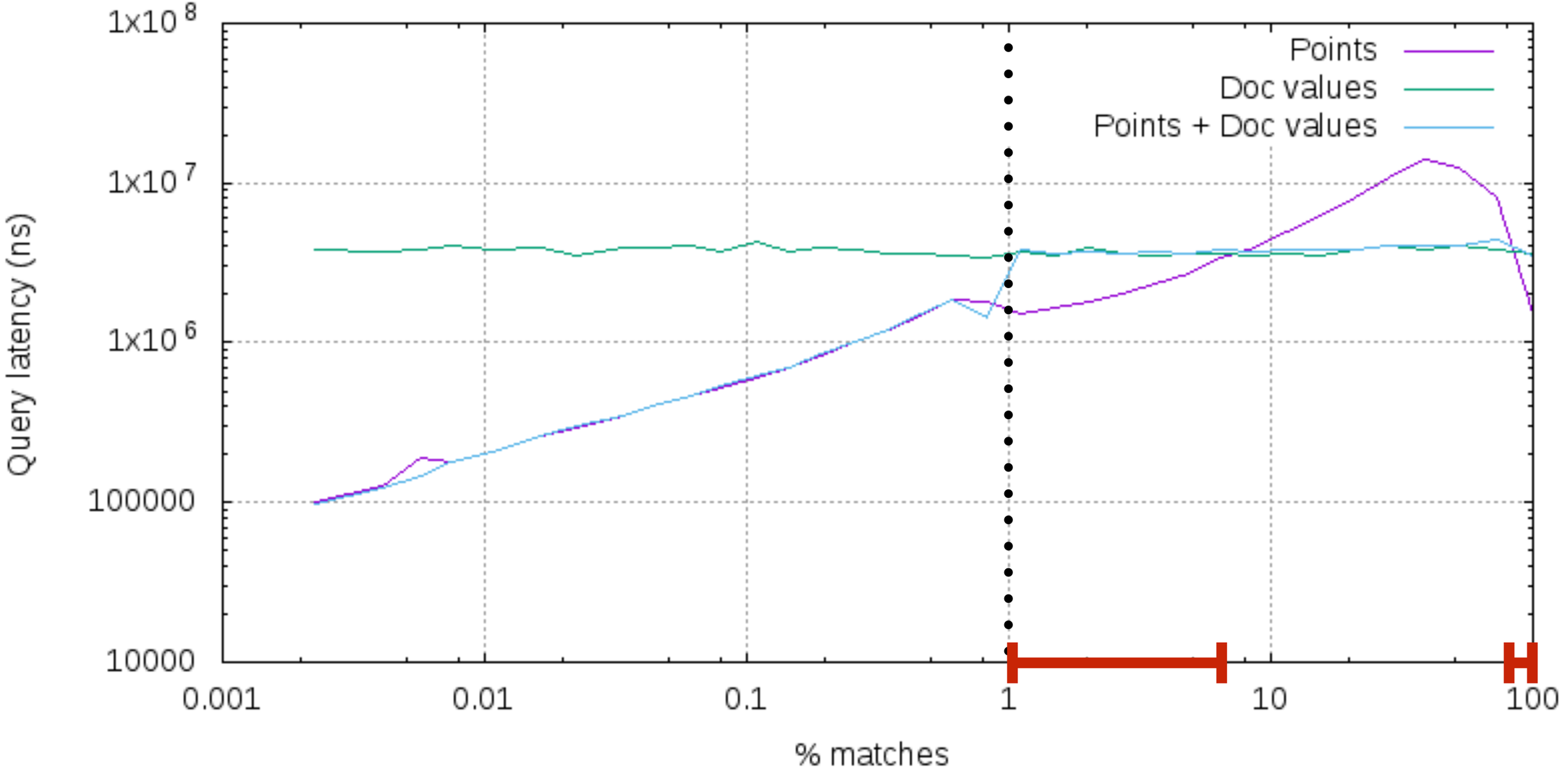
Query planning: benchmark against 0.1% term



Query planning: benchmark against 0.1% term



Query planning: benchmark against 1% term



Query planning: conclusion

- Also works for:
  - geo bounding box queries
  - geo distance queries
- Follow-ups:
  - Improve the heuristics
  - Make it work for prefix / wildcard / fuzzy / terms queries

And more

- Sequence numbers on index operations
- Better query parsing of prefix / wildcard / fuzzy queries
- Boolean similarity
- Unified highlighter
- Optimized geo distance sorting

More Questions?

Visit us at the AMA

www.elastic.co

Please attribute Elastic with a link to elastic.co