

How to offload MySQL server with Sphinx

Vladimir Fedorkov, Sphinx
MySQL & Friends devroom,
FOSDEM, 2012

What is Sphinx

- One more daemon on your box(es)
- Can be queried via
 - API calls (PHP, Python, Java, C, Ruby)
 - MySQL-compatible protocol and SQL queries
 - We are proudly call it SphinxQL
- Never asking riddles

Query example

```
SELECT
  id, user_id, created_time, @weight
FROM
  my_sphinx_index
WHERE
  created_time BETWEEN (X AND Y) AND
  user_id IN (23435, 87945, ... , 23455) AND
  MATCH ('How often is to charge Android phone')
ORDER by @weight DESC, created_time DESC
LIMIT 0, 20
```

Query results sample

```
+-----+-----+-----+-----+
| id      | weight | doc_user_id | created_time |
+-----+-----+-----+-----+
| 7333394 | 1649   | 384139      | 1113235736   |
| 7138085 | 1649   | 402659      | 1113190323   |
| ...     | ...    | ...         | ...          |
| 7051055 | 1649   | 412502      | 1113163490   |
+-----+-----+-----+-----+
20 rows in set (0.00 sec)
```

What's different?

- Meta fields @weight, @group, @count
- No full-text fields in output
 - So far
 - Requires additional lookup to fetch data
- MySQL query become primary key lookup
 - WHERE id IN (33, 9, 12, ..., 17, 5)
 - Good for caching
- Adding nodes is transparent for the application
 - zero downtime or less ;-)

Looking at the Manual

- Integers
 - Int 32bit unsigned (only)
 - Int 64bit signed (only)
 - Set of integers (Multi-Value-Attribute, MVA)
 - Limited ints using bitcount
- Floats
- Strings
- Timestamps

Performance tricks: Geodistance

- Geographical distance is the distance measured along the surface of the earth
 - Two pairs of float values (Latitude, Longitude)
- GEODIST(Lat, Long, Lat2, Long2) in Sphinx

```
SELECT *, GEODIST(docs_lat, doc_long, %d1, %d2) as dist,  
FROM sphinx_index  
ORDER BY dist DESC  
LIMIT 0, 20
```

Performance tricks: Segments

- Grouping results by
 - Price ranges (items, offers)
 - Date range (blog posts and news articles)
 - Ratings (product reviews)
- `INTERVAL(field, x0, x1, ..., xN)`

```
SELECT
    INTERVAL(item_price, 0, 20, 50, 90) as range,
    @count
FROM my_sphinx_products
GROUP BY range
ORDER BY range ASC;
```


Segments: Results example

```
+-----+-----+-----+-----+
| id     | weight | range | @count |
+-----+-----+-----+-----+
| 34545  | 1      | 1     | 654    |
| 75836  | 1      | 2     | 379    |
| 94862  | 1      | 3     | 14     |
+-----+-----+-----+-----+
3 rows in set (0.00 sec)
```

Indexing tricks: sql_joined_field

- Emulates GROUP_CONCAT
- Replaces JOIN while indexing
- Could store several text values from another table into one field.

```
sql_joined_field = dynamic_fields from query; \  
SELECT doc_id, \  
field_value \  
FROM dynamic_fields_values \  
ORDER BY doc_id ASC
```

Performance tricks: MVA

- MVA stands for Multi Value Attributes
 - Array of 32/64bit integers
 - Supported by Real-Time and ondisk indexes
- Useful for shopping categories, page tags, linked documents or items
- Avoiding JOIN on MySQL side

Optimizing database size

- Keep huge text collections out of database.
- `sql_file_field = path_to_text_file`
- `max_file_field_buffer` needs to be set properly

Behind the boolean search

- And, Or
 - hello | world, hello & world
- Not
 - hello -world
- Per-field search
 - @title hello @body world
- Field combination
 - @(title, body) hello world
- Search within first N
 - @body[50] hello
- Phrase search
 - "hello world"
- Per-field weights
- Proximity search
 - "hello world"~10
- Distance support
 - hello NEAR/10 world
- Quorum matching
 - "the world is a wonderful place"/3
- Exact form modifier
 - "raining =cats and =dogs"
- Strict order
- Sentence / Zone / Paragraph
- Custom document weighting
- Different ranking

Performance tricks: Multiqueries

- Common subquery optimization
 - Common part “barack obama” of the queries computed only once:
 - WHERE MATCH('barack obama president')
 - WHERE MATCH('barack obama john mccain')
 - WHERE MATCH('barack obama speech')
 - Subtree cache involved
- Multiqueries are up to 3 times faster
 - reported by our customers on production environment, no synthetic tests performed

Full-text search in non-FT data

- Meta keywords search sometimes faster
 - `__META_AUTHOR_ID_3235`
 - `__META_AUTHOR_NAME_Kelby`
- Use `sql_joined_field`
- Doesn't support ranges

Speeding up high-selective queries

- First letter search
 - __ARTIST_A, __ARTIST_B, __ARTIST_C, ...
- Static ranges emulation with meta_keywords
 - __MY_RANGE_0, __MY_RANGE_1, ...
- Not flexible, but fast

Scale! Scale! Scale!

- Distribute data across the indexes.
 - Better hardware utilization
 - Works in cloud environment
- Combine different indexes
 - Main + Delta
 - On-disk + RT
 - Distributed and local
 - Don't forget about `dist_threads`!
- Use parallel indexing

Where to find more

- <http://sphinxsearch.com/docs>
- «Introduction to search with Sphinx» by O'Reilly
- Invite us to speak on local meetups and conferences
 - Ping me via email vlad@sphinxsearch.com or
twit to [@vfedorkov](https://twitter.com/vfedorkov)
 - Ask me while I'm here.
- Upcoming webinars

Thank you!