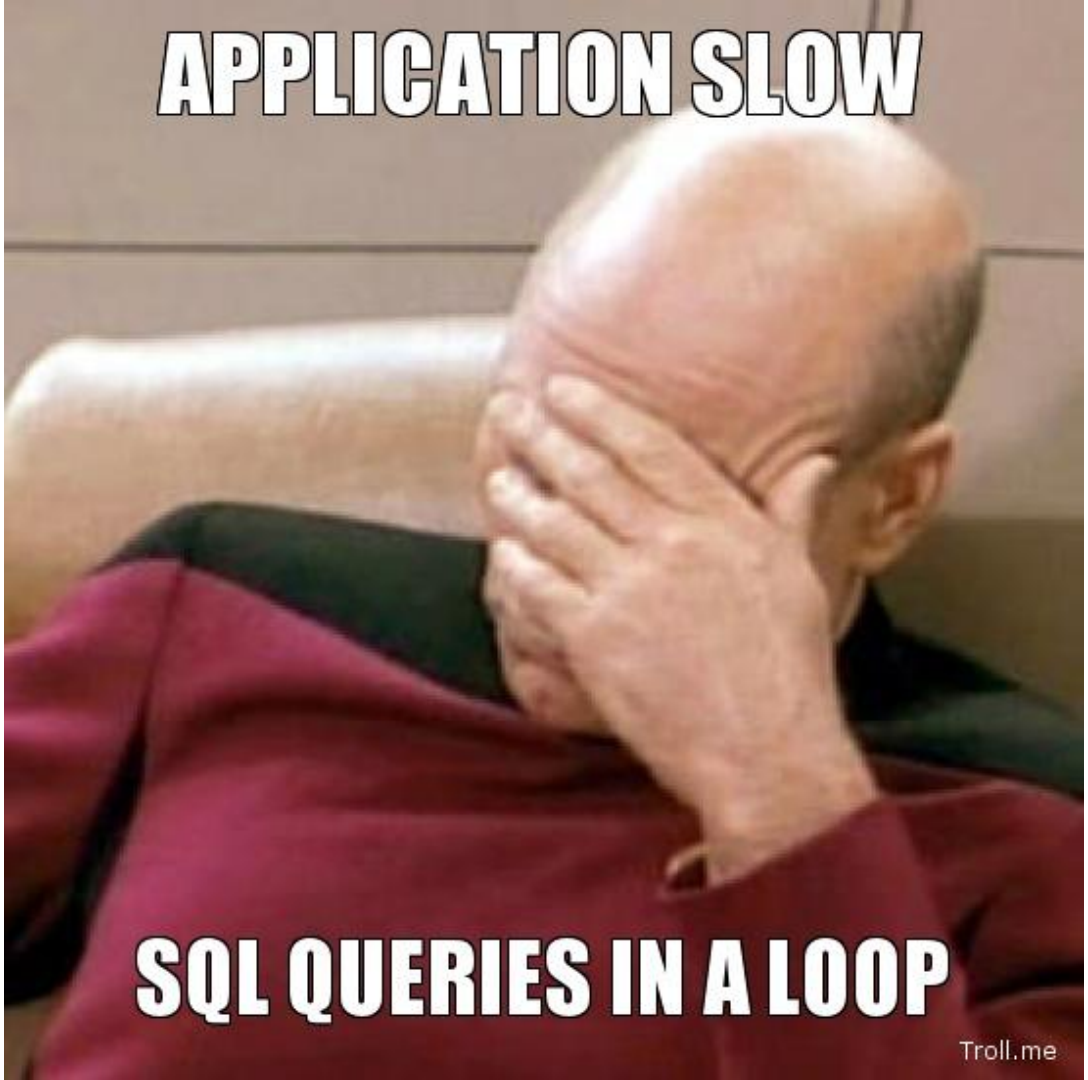
SPHINX SEARCH

Real-Time Index

Sergej Kurakin

Search...


TELL ME MORE
ABOUT YOUR SUPER SEARCH
WITH LIKE %...%
makeameme.org


APPLICATION SLOW
SQL QUERIES IN A LOOP
Troll.me


JOINS JOINS JOINS
JOINS EVERYWHERE
makeameme.org

Search Using only MySQL

- LIKE %keyword%
- FULLTEXT
- JOIN on JOIN on JOIN and then GROUP BY
- LOOPing through data
- Filesort kills performance

Search Example

SELECT * FROM users
LEFT JOIN products ...
LEFT JOIN product_categories …
LEFT JOIN ads …
LEFT JOIN ads_categories…
WHERE ... LIKE ... LIKE ... LIKE ... LIKE …
GROUP BY … ORDER BY ...

But we have Sphinx!

About Sphinx

- http://sphinxsearch.com/
- Fulltext Search engine
- Open Source
- GPL License
- Was presented by Vaidas Žilionis at VilniusPHP at 2013-01-03
- You can find more on Internet


CVN 74
USS JOHN C STENNIS

Recommended solution by Percona

[High Performance MySQL]

Ordinary Stuff

Disk Indexes (pros)

- maximum indexing speed
- searching speed
- keeping the RAM footprint as low as possible

Disk Indexes (cons)

- cost of text index updates
- rebuild the entire index from scratch

Real Life Situation

Every moment

- Customers submit data into your database
- New products added to database
- Products updated in database

When search index will be updated???


DEEP TROUBLE

Off topic: Psychology

Some smart people say it’s bad to say troubles.

We should convert troubles to challenges and solve them.

Lets believe them.


TROUBLES? WE CALL IT
CHALLENGE
makeameme.org

The Challenge

- Move search from slow MySQL
- Update Search Index in near Real-Time
- Test and adopt new technology
- Minimize support price
- Support of Lithuanian language

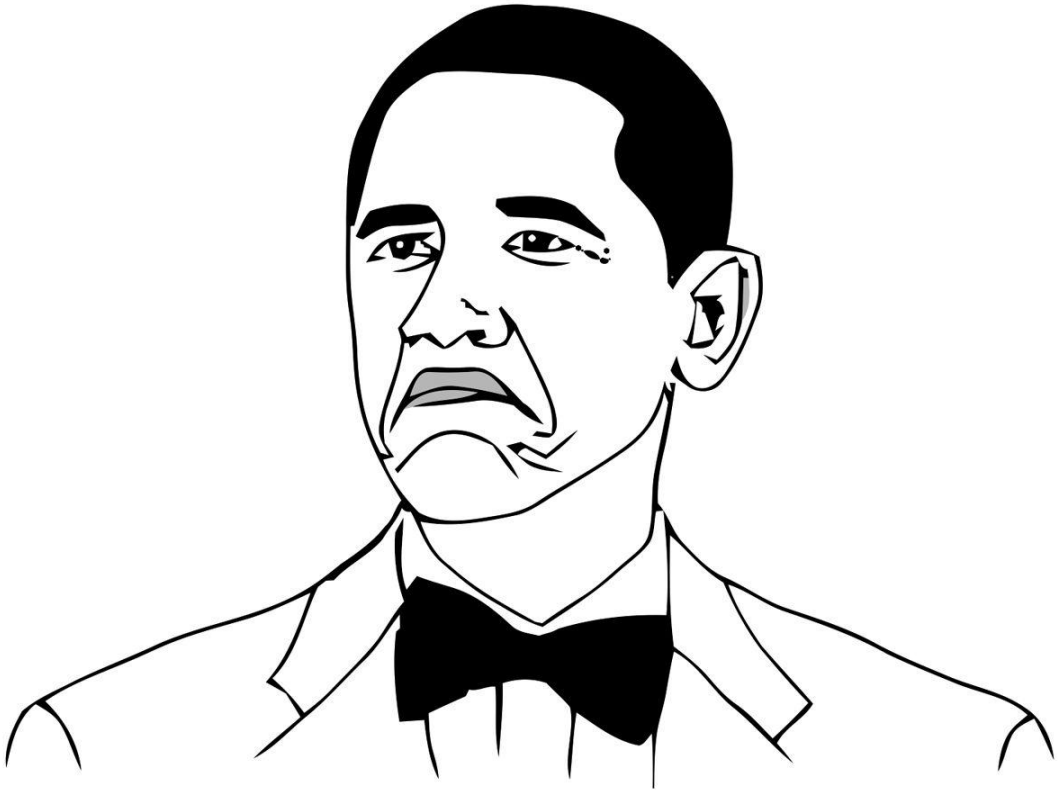
Sphinx has Real-Time Indexes

Real-Time Indexes (pros)

- dynamic updates
- incremental additions
- "soft real-time" in terms of writes

Real-Time Indexes (cons)

- larger memory footprint
- index updates may be late for a second
- index updates can be done using only SphinxQL



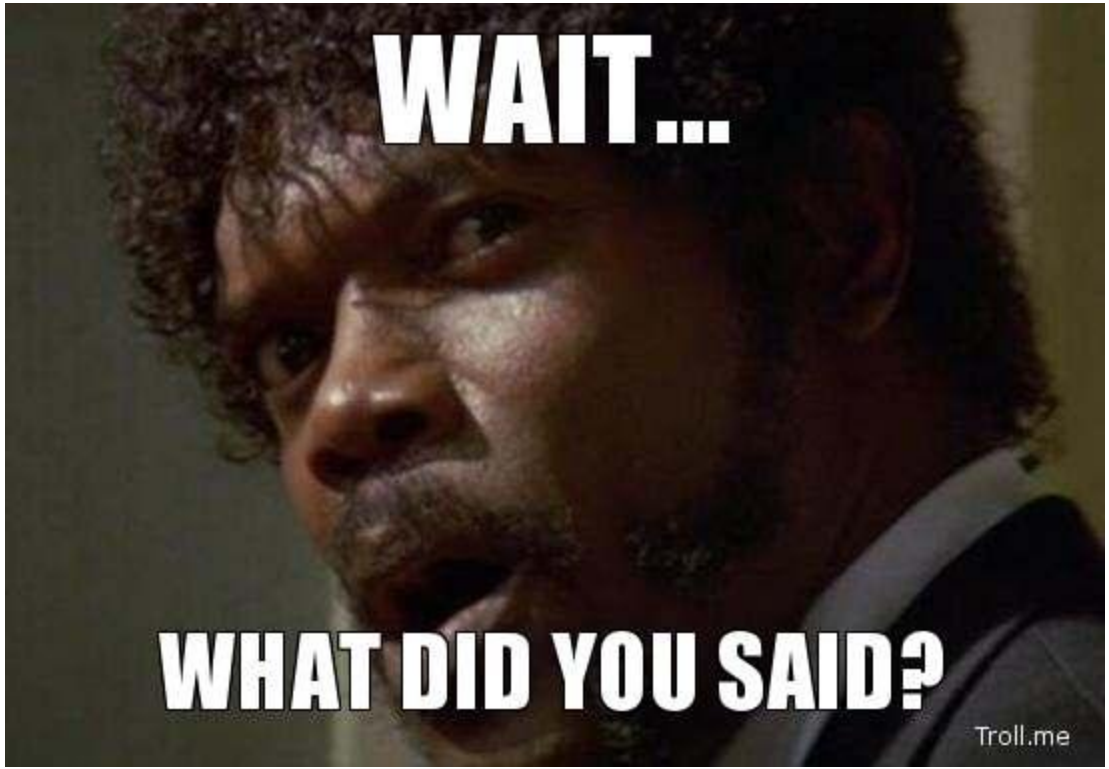
NOT BAD

Real-Time Index (requirements)

- data sources are not required and ignored
- explicitly enumerate all the text fields, not just attributes

Real-Time Index Declaration

```
index rt {
   type = rt
   path = /var/lib/sphinx/data/rt
   rt_field = title
   rt_field = content
   rt_attr_uint = gid
}
```


WAIT...
WHAT DID YOU SAID?
Troll.me

SphinxQL

About SphinxQL (from Manual)

SphinxQL is our SQL dialect that exposes all of the search daemon functionality using a standard SQL syntax with a few Sphinx-specific extensions. Everything available via the SphinxAPI is also available via SphinxQL but not vice versa; for instance, writes into RT indexes are only available via SphinxQL.

Supported Statements

SELECT, INSERT, REPLACE, UPDATE, DELETE

BEGIN, COMMIT, ROLLBACK

OPTIMIZE INDEX, SHOW STATUS, SET, SHOW TABLES, DESCRIBE, ALTER

SELECT

- Syntax is based upon regular SQL
- Currently missing support for JOINs
- Several Sphinx-specific extensions

INSERT

- Only supported for Real-Time indexes
- ID column must be present in all cases
- Expressions are not currently supported

REPLACE

- Identical to INSERT.

Note: Rows with duplicate IDs will not be overwritten by INSERT; use REPLACE to do that

UPDATE

- Real-Time and disk indexes are supported
- WHERE has the same syntax as in the SELECT

DELETE

- Only supported for Real-Time indexes
- WHERE has the same syntax as in the SELECT

Transactions

Transactions

- BEGIN statement forcibly commits pending transaction
- Transactions are limited to a single RT index
- Transactions are limited in size
- Overly isolated (same session isolation)

OPTIMIZE INDEX

- Real-Time index optimization in a background thread
- No way to check the index or queue fragmentation status
- Needs to be issued manually

ALTER

- Supports adding and dropping of one attribute at a time
- Limit on attribute type
- Querying of an index is impossible while adding a column
- Won’t work on indexes without any attributes
- Be really careful

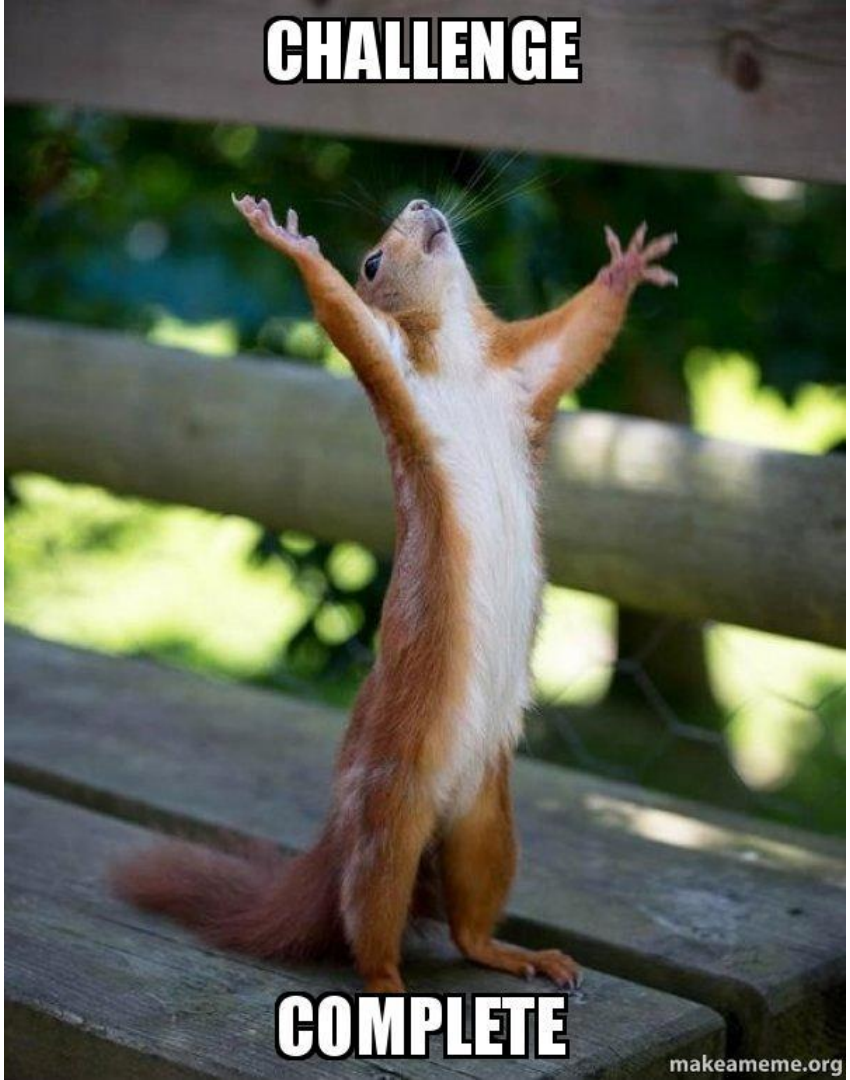
SphinxQL

- Based on MySQL API (mysql41)
- Listens on TCP port 9306
- You can always connect with MySQL client
- Better use simple queries


COOKING
KISS
LIKE A BOSS
quickmeme.com

The Data

- Prepare/pre-process before putting to Sphinx
- Use background jobs if possible (Job Queues)
- Be ready for full reindex


CHALLENGE
COMPLETE
makeameme.org

Challenge Complete

- Move search from slow MySQL
- Update Search Index in near Real-Time
- Test and adopt new technology
- Minimize support price
- Support of Lithuanian language

Questions?

Sergej Kurakin

Work Email: sergej.kurakin@nfq.lt
Personal Email: sergej@kurakin.info
https://www.linkedin.com/in/sergejkurakin

Special thanks to authors of all pictures used in this presentation.