Advanced search capabilities with MySQL and Sphinx

Vladimir Fedorkov, Blackbird

Andrew Aksyonoff, Sphinx

Percona Live MySQL UC, 2014

Knock knock who's there

- Vladimir
  - Used Sphinx in production since 2006
  - Performance geek
  - Blog http://astellar.com, twitter @vfedorkov
  - Works for Blackbird
- Andrew
  - Created Sphinx, http://sphinxsearch.com
  - Just some random guy

Search is important

- This is 2014, Google spoiled everyone!
- Search needs to exist
- Search needs to be fast
- Search needs to be relevant

- Today, we aim to show you how to start
  - With Sphinx, obviously

Available solutions

- Most databases have integrated FT engines
  - MySQL (My and Inno), Postgres, MS SQL, Oracle…
- Standalone solutions
  - Sphinx
  - Lucene / Solr
  - Lucene / ElasticSearch
- Hosted services
  - IndexDen, SearchBox, Flying Sphinx, WebSolr, …

Why Sphinx?

- Built-in DB search sucks

- Sphinx works great with DBs and MySQL
- Sphinx talks SQL => zero learning curive
- Fast, scalable, relevant, and other buzzwords :P

- You probably heard about Lucene anyway
- NEED MOAR DIVERSITY

What Sphinx is not

- Not a plugin to MySQL
- Does not require MySQL
- Not SQL-based (but we talk SQL)
  - Non-SQL APIs are available
- Not a complete database replacement
  - Yet?
  - Ever! OLAP vs OLTP vs Column vs FTS vs Webscale

Quick overview

- Sphinx = standalone, open-source **search server**
- Supports **Real-time** indexes
- **Fast**
  - 10+ MB/sec/core indexing, 700+ qps/core searching
  - And counting!
- **Scalable**
  - Can do a lot even on 1 box
  - Lets you aggregate search results from N boxes
  - Auto-sharding, replication etc in the works
- **Easy** to integrate, via **SphinxQL**

Sphinx records

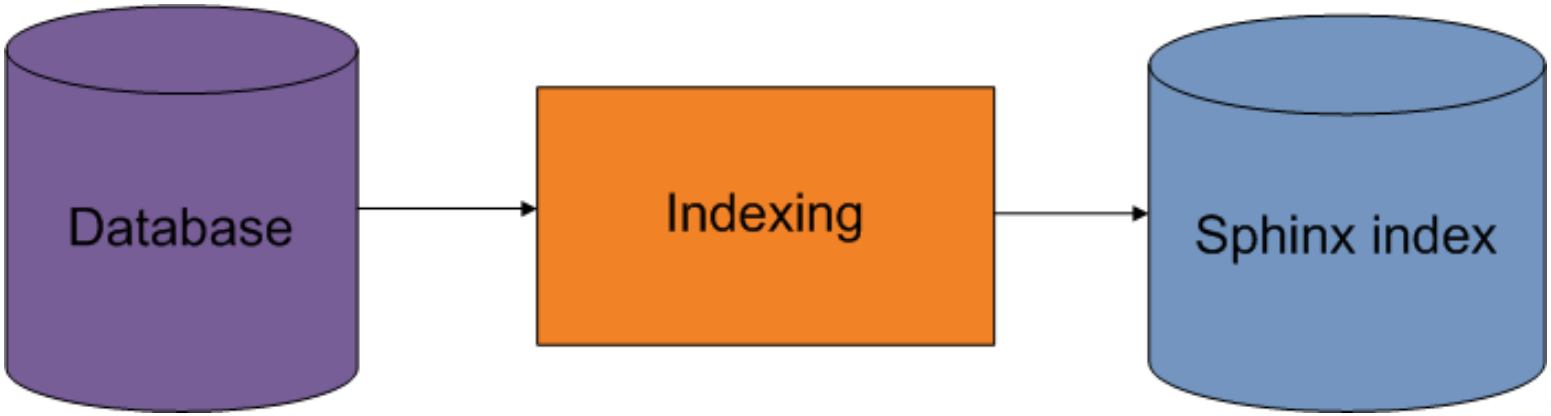
- 10,000,000,000s of documents
  - 30,000,000,000+ docs at Infegy, BoardReader etc
  - 1200+ servers, secret client
- 300,000,000+ queries per day (craigslist)
- 10-1000x faster than MySQL on text searches
  - How come? Designed for search, not OLTP
  - In-memory attributes, fixed memory target, etc

Installation

- From binary packages
  - http://sphinxsearch.com/downloads/
  - Builds for RedHat, Ubuntu, MacOS, Windows, …
- From source
  - http://sphinxsearch.googlecode.com/svn/
  - ./configure && make && make install

Initial configuration: indexing

- Config says…
  - Where to look for data?
  - How to process it?
  - Where to store the index?



Where to look for the data?

- MySQL
- PostgreSQL
- MSSQL
- Oracle
- ODBC source
- XML pipe
- CSV pipe
- …

MySQL source

```
source data_source
{
        …
        sql_query = \
                SELECT id, channel_id, ts, title, content \
                FROM mytable

        sql_attr_uint           = channel_id
        sql_attr_timestamp      = ts
        …
}
```

A complete version

```
source data_source
{
  type         = mysql
  sql_host     = localhost
  sql_user     = myuser
  sql_pass     = mybiggestsecret
  sql_db       = test

  sql_query_pre = SET NAMES utf8
  sql_query     = SELECT id, channel_id, ts, title, content \
                  FROM mytable WHERE id>=$start and id<=$end
  sql_attr_uint = channel_id
  sql_attr_uint = ts

  sql_query_range  = SELECT MIN(id),  MAX(id) FROM mytable
  sql_range_step   = 1000
}
```

How to process the data: Index config

```
index my_sphinx_index
{
      source                  = data_source
      path                    = /my/index/path/idx

      html_strip              = 1
      morphology              = lemmatize_en_all
      stopwords               = stopwords.txt
}
```

Indexer configuration

```
indexer
{
    # upto 2047M
    mem_limit   = 512M

    # for a busy system
    max_iops    = 40
    max_iosize = 1048576
}
```

Running indexer

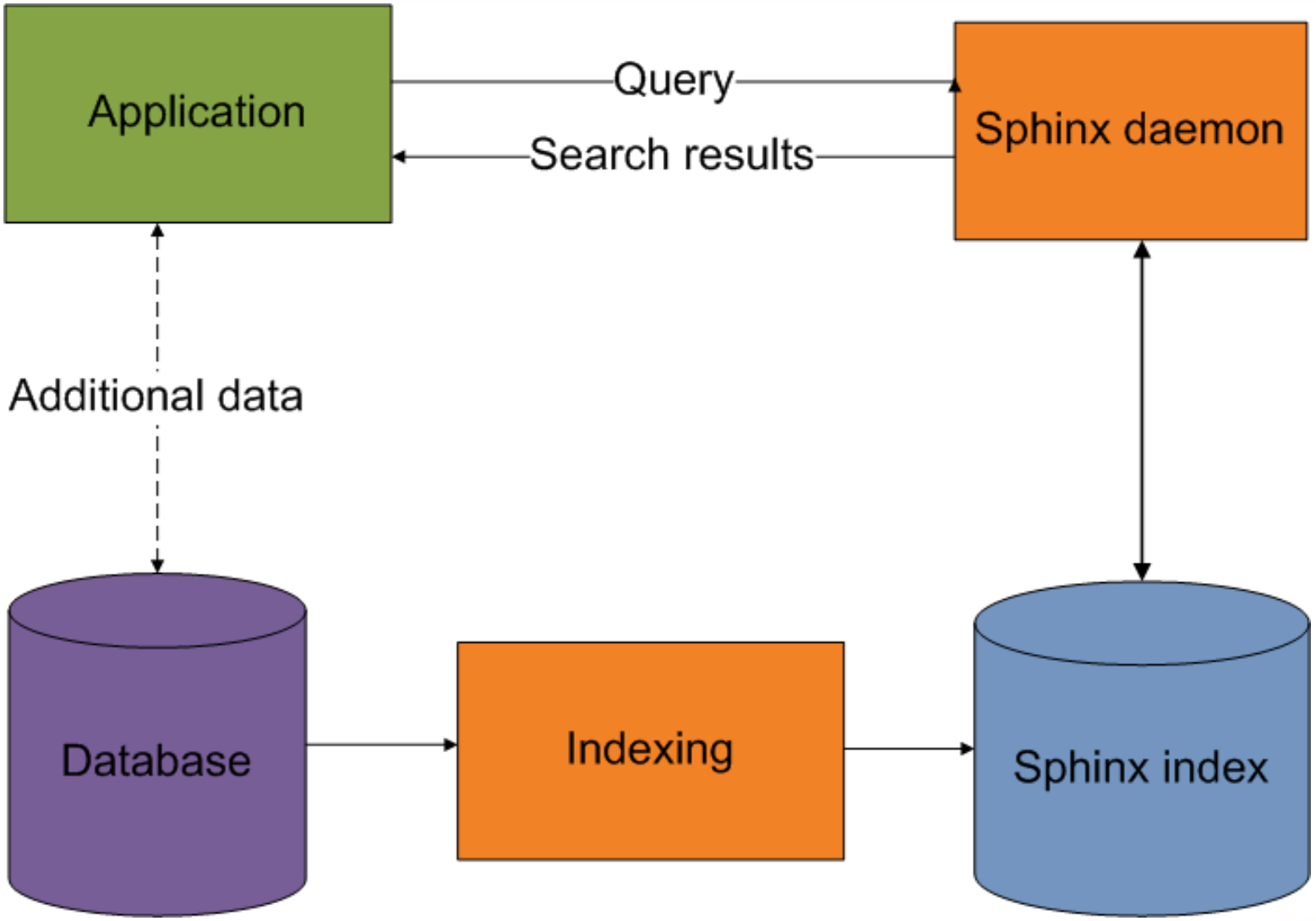
```
$ ./indexer my_sphinx_index
Sphinx 2.0.2-dev (r2824)
Copyright (c) 2001-2010, Andrew Aksyonoff
Copyright (c) 2008-2010, Sphinx Technologies Inc (http://sph...

using config file './sphinx.conf'...
indexing index 'my_sphinx_index'...
collected 999944 docs, 1318.1 MB
sorted 224.2 Mhits, 100.0% done
total 999944 docs, 1318101119 bytes
total 158.080 sec, 8338160 bytes/sec, 6325.53 docs/sec
total 33 reads, 4.671 sec, 17032.9 kb/call avg, 141.5 msec/call
total 361 writes, 20.889 sec, 3566.1 kb/call avg, 57.8 msec/call
```

ToDo

- Run the daemon
- Connect, send search query
- Get search results to the application
- Maybe get extra stuff (query statistics, build snippets, log the performance profile, etc)

- (brace yourself for a meaningless diagram)

How shit works



Configuring searchd

```
searchd
{
    listen = localhost:9312
    listen = localhost:9306:mysql4

    query_log           = query.log
    query_log_format    = sphinxql

    pid_file            = searchd.pid
}
```

Connecting to Sphinx…

- Sphinx API
  - PHP, Python, Java, Ruby, C is included in distro
  - .NET, Rails (via Thinking Sphinx) via third party libs

- Or, better yet, **SphinxQL**
  - MySQL-compatible protocol
  - Good for you, good for me

…Just like MySQL

```
$ mysql -h 127.0.0.1 -P 9306
Welcome to the MySQL monitor.  Commands
end with ; or \g.
Your MySQL connection id is 1
Server version: 2.1.0-id64-dev (r3028)


Type 'help;' or '\h' for help. Type '\c'
to clear the current input statement.


mysql>
```

SphinxQL. Search against 8M rows.

```
mysql> SELECT id, ...
    -> FROM myisam_table
    -> WHERE MATCH(title, content_ft)
    -> AGAINST ('I love sphinx') LIMIT 10;
...
10 rows in set (1.18 sec)
```

MySQL

```
mysql> SELECT * FROM sphinx_index
        -> WHERE MATCH('I love Sphinx') LIMIT 10;
...
10 rows in set (0.05 sec)
```

Sphinx

SphinxQL

- MySQL protocol, SQL syntax, own dialect
- Works ***without*** MySQL
  - MySQL client library still required of course
- The only interface to RT indexes
- Same same (to MySQL) but different
  - WHERE MATCH vs WHERE MATCH AGAINST
  - GROUP <N> BY
  - Implicit, ever present LIMIT
  - etc etc etc

…But not quite!

```
mysql> SELECT * FROM idx
    -> WHERE MATCH('I love Sphinx') LIMIT 5
    -> OPTION field_weights=(title=100, content=1);
+---------+--------+------------+------------+
| id      | weight | channel_id | ts         |
+---------+--------+------------+------------+
| 7637682 | 101652 |     358842 | 1112905663 |
| 6598265 | 101612 |     454928 | 1102858275 |
| 6941386 | 101612 |     424983 | 1076253605 |
| 6913297 | 101584 |     419235 | 1087685912 |
| 7139957 |   1667 |     403287 | 1078242789 |
+---------+--------+------------+------------+
5 rows in set (0.00 sec)
```

How SphinxQL is different from regular SQL?

- Own generic extensions like
  - WITHIN GROUP ORDER BY
  - GROUP <N> BY etc
- Own search related extensions like
  - OPTION for fine grained search query control
  - SHOW META for query information
  - CALL SNIPPETS for snippets
  - CALL KEYWORDS for keyword info, stats

Group by example

```
mysql> SELECT id, WEIGHT(), ts, YEAR(ts), COUNT(*) as yr
    -> FROM lj1m WHERE MATCH('I love Sphinx')
    -> GROUP BY yr ORDER BY yr DESC LIMIT 5
    -> OPTION field_weights=(title=100, content=1);
+---------+----------+------------+------+----------+
| id      | weight() | ts         | yr   | count(*) |
+---------+----------+------------+------+----------+
| 7637682 |   101652 | 1112905663 | 2005 |       14 |
| 6598265 |   101612 | 1102858275 | 2004 |       27 |
| 7139960 |     1642 | 1070220903 | 2003 |        8 |
| 5340114 |     1612 | 1020213442 | 2002 |        1 |
| 5744405 |     1588 |  995415111 | 2001 |        1 |
+---------+----------+------------+------+----------+
```

Advanced search techniques

- Full-text matching
- Non-text filtering, grouping, sorting
  - Offloading MySQL as well
- Faceted search
  - A special case of grouping
- Other search-based services
  - Relevancy, typos, suggestions, etc

Full-Text matching operators

- 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
- Document structure support
  - Sentence
  - Zone
  - Paragraph

Non-text filtering

- Or in SphinxQL terms, WHERE conditions
  - a = 5, a < 5, a > 5, a BETWEEN 3 AND 5
- Or in SphinxAPI call terms, filters
  - SetFilter(), SetFilterRange(), SetFilterFloatRange()
- Works on non-text attributes
- Applied ***after*** full-text condition from MATCH()

Non-text attribute types

- WHERE, ORDER, GROUP… not just for ints
- Integers, floats, strings
  - 32 bit unsigned, 64 bit signed, and bitfields
- MVAs (essentially sets of 32 bit integers)
- JSON!
  - SELECT ALL(x>3 AND x<7 FOR x IN j.intarray)
  - SELECT j.users[3].address[2].streetname
  - Efficient packed format internally, “SphinxBSON”

Quality search takes more than keywords

- Relevance tuning
- Drill-down (narrowed search, faceted search)
- Query rewriting
- Typo corrections
- Search query autocompletion
- Related documents
- …
- A topic worth its own book. Or three.

Search by GEO-Distance

- Bumping up and/or filtering local results
  - Just add float latitude, longitude attributes, and..
- **GEODIST**(Lat, Long, Lat2, Long2) in Sphinx
- Now has syntax for mi/km/m, deg/rad etc

SELECT *, **GEODIST(doc_lat, doc_long, $lat, $long)** as dist,
FROM sphinx_index ORDER BY dist DESC LIMIT 0, 20

Search within range

- Price ranges (items, offers)
- Date range (blog posts and news articles)
- Ratings, review points
- 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

Query rewriting

- THE easiest search quality trick?
- If it returns 0 results, REWRITE IT
  - Relax the “match phrase” if any
  - Relax the “match all words” if any
  - Switch to /0.3 quorum
  - Switch to match any ie. /1 quorum

Spelling corrections

- Basically a “Did you mean”
- Fix the typos
  - Britney vs Brittney vs Brittaney…
- Fix the keyboard layout issues
  (eg. combined ru/en layout)
- Usually needs to be trained on ***your*** database
  - Camera vs Camara vs Camaro..

Sphinx vs spelling corrections

- Demo PHP script, /misc/suggest/
  - suggest.conf defines an auxiliary Sphinx index
  - suggest.php is an actual implementation
- Trigrams for initial matching
- Then reranking best matches by Levenstein()

- Just a demo: not complete, not tuned to your thresholds, might need UTF8 fixes, etc…

Sphinx vs Related search implementation

- Just use the very same Sphinx index
- Start with a quorum operator, sort by relevance
  - “Sony NEX-5N”/0.3
  - “Romney wonders why airplane windows don’t open”/2
- Improve on that using custom sorting
  - Same category => boost weight
  - Close price => boost weight
  - Same vendor => boost weight
  - Same car year in car category only => boost weight

Excerpts (snippets)

- **CALL SNIPPETS** (or BuildExcerpts in the API)
- Options
  - before_match ""
  - after_match ""
  - chunk_separator "…"
  - limit
  - around
  - force_all_words
  - …and a bunch more

Relevance tuning

- Sphinx has ***expression based ranking***
- 15+ of our text signals, N of yours non-text
  - OPTION ranker=expr('1000*sum(lcs)+bm25')
  - OPTION ranker=expr('700*sum(lcs)+bm25f(1.4, 0.8, {title=3, content=1}')
  - OPTION ranker=expr('$A*sum(lcs) + $B*atc + $C*bm25f + $D*…')

Relevance tuning

- Human judgments (assessments)
- Automate early, 10 queries is already it
- Don’t be afraid, 1000+ judgments/day is easy
- Compute metrics, many metrics
  - P, R, MAP or NDCG, J, …
- Compare metrics, perhaps optimize on them

Instead of conclusion… we haven't even mentioned:

- Real time indexes (soft real time)
- Advanced text processing (morphology, blend_chars, filter plugins, etc)
- Geosearching, facets, etc (yes we can)
- Cluster architectures, HA/LB advice
- …

- So go try Sphinx, its pretty cool ;)

Questions!

Companies:

Blackbird: http://blackbird.io

Sphinx: http://sphinxsearch.com

Follow speakers on twitter:

Vladimir: @vfedorkov

Andrew: @shodanium

Thank you!

Visit our booths in expo hall and our websites:

Blackbird: http://blackbird.io

Sphinx: http://sphinxsearch.com

We're all hiring!!! ☺

Thank you!