



PERCONA
Performance Consulting Experts

Using Sphinx Beyond Full Text Search

Apr 11-14, 2011

MySQL Conference and Expo

Santa Clara, CA

by Peter Zaitsev, Percona Inc,

Vladimir Fedorkov, Sphinx
Technologies Inc

The Sphinx

- Free open source search server
- Begins 10 years ago as a full text daemon
- Now powerful, fast, relevant, scalable search engine.
- Not only Full Text search!

Sphinx installations

- Over 16,000,000,000 (yes billions) documents
 - boardreader.com, over 5Tb data on about 40 boxes
- Over 200,000,000 queries/day (craigslist.org)
 - 2,000 QPS against 15 Sphinx boxes
- Also powers NetLog, Meetup, Slashdot, WikiMapia, and a few thousands other sites
 - <http://sphinxsearch.com/info/powered/>

Why Sphinx?

- Extended Full-text query syntax
 - & | () @field, phrase, proximity, NEAR/x, quorum, ...
- You can run it on various platforms
 - Even AIX and iPhone!
 - PHP, Python, Java, Ruby, C binary protocol APIs are officially available.
 - .NET, Thinking Sphinx (for Rails) and few more available as third party plugins
- Even better – now we have SphinxQL

SphinxQL?

- Our own implementation of MySQL protocol
- Our own SQL parser
- MySQL not required!
- Any client library (eg. PHP's or .NET) should suffice
- All new features will initially appear in SphinxQL

SphinxQL!

```
$ mysql -h 0 -P 9306
```

```
Welcome to the MySQL monitor.  Commands end with ; or \g.
```

```
Your MySQL connection id is 1
```

```
Server version: 1.11-dev (r2569)
```

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

```
mysql> SELECT * FROM ljl_m WHERE MATCH('Sphinx') ORDER BY ts DESC LIMIT 3;
```

```
+-----+-----+-----+-----+
| id      | weight | channel_id | ts          |
+-----+-----+-----+-----+
| 7333394 | 1649   | 384139     | 1113235736 |
| 7138085 | 1649   | 402659     | 1113190323 |
| 7051055 | 1649   | 412502     | 1113163490 |
+-----+-----+-----+-----+
```

```
3 rows in set (0.00 sec)
```

Why Sphinx?

- 10-1000x vs MySQL on full-text searches
 - MySQL only behaves when indexes are in RAM
- 2-3x vs MySQL on non-full-text scans
 - Grouping and sorting in fixed memory
 - Attribute search block skipping
- Fast indexing
 - Up to 10Mb/s on a single core on the home desktop.

Query speed 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> SELECT * FROM sphinx_index  
-> WHERE MATCH('I love Sphinx') LIMIT 10;
```

...

```
10 rows in set (0.05 sec)
```

Indexing speed

```
$ ./indexer lj1m
Sphinx 1.11-dev (r2569)
Copyright (c) 2001-2010, Andrew Aksyonoff
Copyright (c) 2008-2010, Sphinx Technologies Inc (http://sph...

using config file './sphinx.conf'...
indexing index 'lj1m'...
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
```

Not only Full-Text!

```
mysql> SELECT YEARMONTH(ts) yyyyymm, COUNT(*) cnt
-> FROM sphinx_index GROUP BY yyyyymm
-> ORDER BY yyyyymm DESC LIMIT 10;
```

id	weight	yyyyymm	@count	@groupby	@count
----	--------	---------	--------	----------	--------

...

```
10 rows in set (1.39 sec)
```

```
mysql> SELECT DATE_FORMAT(ts, '%Y%m') as yyyyymm, COUNT(*)
-> FROM innodb_table
-> GROUP BY yyyyymm ORDER BY yyyyymm DESC LIMIT 10 ;
```

yyyyymm	COUNT(*)
---------	----------

...

```
10 rows in set (2 min 53.54 sec)
```

Available data sources

- MySQL
- PostgreSQL
- MSSQL
- ODBC source
- XML pipe
 - Also useful for custom documents pre-processing

Multi-valued attribute (MVA)

- Several values attached to the document
 - Designed for 1:M relations
- Useful for
 - Page tags
 - Item belongs to several categories
- SQL join optimization
 - Avoid joins at all
 - As simple as
sql_joined_field = tags from query;
SELECT docid, CONCAT('tag',tagid)
FROM tags ORDER BY docid ASC

MVA in action

```
mysql> SELECT mva_field FROM sphinx_index \  
-> WHERE MATCH('test') AND mva_field IN (1,2,3,4) LIMIT 1;  
-> SHOW META;
```

```
+-----+-----+-----+  
| id      | weight | mva_field|  
+-----+-----+-----+  
| 20034267 | 4647 | 1      |  
+-----+-----+-----+  
1 row in set (0.05 sec)
```

```
+-----+-----+  
| Variable_name | Value |  
+-----+-----+  
| total         | 1000 |  
| total_found   | 29925 |  
| time          | 0.057 |  
| keyword[0]    | test |  
| docs[0]       | 30590 |  
| hits[0]       | 61719 |  
+-----+-----+  
6 rows in set (0.01 sec)
```

Geo distance search

- A pair of float attributes
 - In radians
- Can be used in sorting
- “between” is also available
- GEODIST(lat1,long1,lat2,long2) is available in SphinxQL
 - returns results in meters

Geodistance in action

```
mysql> SELECT location_id, latitude, longitude,  
-> GEODIST(latitude, longitude, 0.651137, -2.127562) as geodist  
-> FROM sphinx_index ORDER BY geodist ASC LIMIT 10;
```

id	weight	location_id	longitude	latitude	geodist
81875993	1	16316	-2.127562	0.651137	2.859948
81875994	1	16316	-2.127562	0.651137	2.859948
81875996	1	16316	-2.127562	0.651137	2.859948
81875997	1	16316	-2.127562	0.651137	2.859948
81875999	1	16316	-2.127562	0.651137	2.859948
81876000	1	16316	-2.127562	0.651137	2.859948
81876001	1	16316	-2.127562	0.651137	2.859948
81876002	1	16316	-2.127562	0.651137	2.859948
81876003	1	16316	-2.127562	0.651137	2.859948
81876004	1	16316	-2.127562	0.651137	2.859948

```
10 rows in set (0.20 sec)
```

```
mysql>
```

Unix timestamps

- UNIX timestamp basically
 - `sql_attr_timestamp = added_ts`
- Time segments + relevance sorting is available
 - results would change over time
- Time fragmentation
 - last hour/day/week/month/3 months
 - everything else
- Grouping by time segments are available

Numeric attributes

- Integer
 - sql_attr_uint
 - 32bit unsigned, a simple integer value.
- Bigint
 - sql_attr_bigint
 - 64-bit signed integer
 - Available for mysql, pgsql, mssql sources only
- Floating point attributes
 - sql_attr_float
 - Single precision, 32-bit IEEE 754 format
- Just like in MySQL

Non numeric attributes

- **String attributes**
 - `sql_attr_string`
 - Not included into full-text index, stored in memory
 - Available since 1.10-beta
- **Wordcount attribute**
 - `sql_attr_str2wordcount`
 - A separate attribute that counts number of words inside the document
 - mysql, pgsql, mssql sources only
 - Since 1.10-beta

File field

- `sql_file_field = <path_column_name>`
- Reads document contents from file system instead of database.
 - Offloads database
 - Prevents cache trashing on database side
 - Much faster in some cases
- mysql, postgres, mssql sources only
- Since 1.10-beta

Faceted searching support

- Based on multiqueries
 - Same Full-text parts with different GROUP BY and sort by parts.
- Grouping queries
 - i.e. top X authors, top Y vendors
- Sorting by another field
 - ASC, DESC

Faceted search in action

```
mysql> SELECT *, YEARMONTH(ts) yyyyymm FROM lj1m  
-> WHERE MATCH('my search') LIMIT 2;
```

id	weight	channel_id	ts	yyyyymm
6274606	2635	475928	1084998805	200405
6814005	2626	433879	1112335196	200504

2 rows in set (0.00 sec)

```
mysql> SELECT YEARMONTH(ts) yyyyymm, COUNT(*) cnt  
-> FROM lj1m WHERE MATCH('my search') GROUP BY yyyyymm  
-> ORDER BY yyyyymm DESC LIMIT 4;
```

id	weight	yyyyymm	@count
6814005	2626	200504	1289
6034886	2616	200503	2658
5416370	2594	200502	1596
5154547	2608	200501	1195

4 rows in set (0.00 sec)

Multiquery performance

- 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 3x faster
 - reported by our customers on production environment, no synthetic tests performed

Typical Sphinx applications

- Shopping items and goods search
- Forums & blogs search
- Data mining application
- News search
- Search against torrents list of files
 - Prefix & infix search in action
- Dating websites
- Local content search
 - Embedded Sphinx

Shopping items search

- Geo search
 - For store location
- MVA
 - Categories, seller, etc
- Custom weighting
- Full Text search by
 - title
 - product description
 - against customer review
- Sorting by popularity, price, distance

Forum/blog/news search

- Time segment sorting
 - Especially for the news
- MVA for user tags
- Category filtering
- Easy migration from MyISAM

Dating websites

- Full-scan oriented queries
 - Search by age, sex, location
- Data can be scaled well
 - Almost linear dependence between number of cores and queries speed
 - Because of fast in-memory lookups

Typical services

- "Similar items/pages" service
 - Using quorum & custom weighting
 - Can do news aggregation with some tuning
- Misspelling correction service
 - By external script (included in distribution)

Built in optimizations

- Non-related filter blocks are skipped during lookups
- All non full-text attributes stored in memory
 - No slow disk IO operations at all
 - Be careful with string attributes!
- Highly scalable using distributed indexes
 - Sharding data across 5 servers will gives your about 5 times faster search

Distributed indexes

```
index distributed_sphinx_index1
{
    type          = distributed
    local         = ondisk_index1
    local         = ondisk_index2
    local         = ondisk_index3
    local         = ondisk_index4
}
...
dist_threads = 4
...
```

Distributed indexes explained

- Query a few indexes on the same box
 - `dist_threads` option tell Sphinx how many cores to use for the single query
- Query indexes across the servers
 - Transparent for application
 - Master node performs only aggregation
 - Can be combined with local indexes on the same box!

Now Real-time!

- Push model instead of Pull for on-disk indexes
 - via INSERT/UPDATE/DELETE
- Update data on the fly
- Formally "soft-realtime"
 - As in, most of the writes are very quick
 - But, not guaranteed to complete in fixed time
- Transparent for application
- Can be combined with non-RT indexes
 - Even running on different boxes

Combining on-disk and RT indexes

```
index distributed_sphinx_index2
{
  type      = distributed
  agent     = 192.168.100.51:9312:ondisk_index1
  agent     = 192.168.100.52:9312:ondisk_index2
  agent     = 192.168.100.53:9312:rt_index
}
```

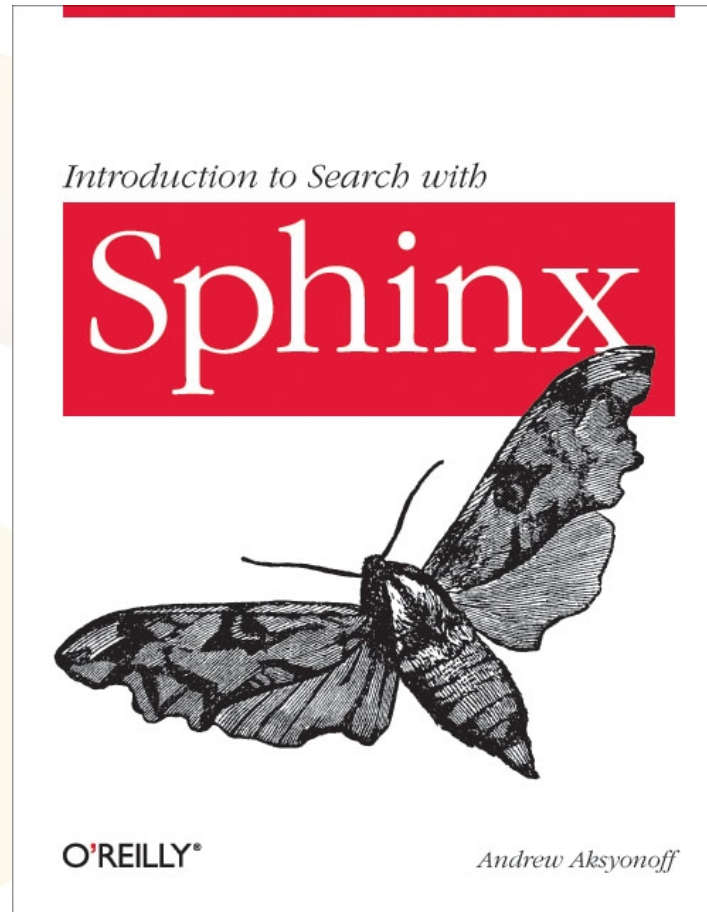
RT indexes, the differences

- Indexing is SphinxQL only
 - mysql_connect() to Sphinx instead of MySQL
 - mysql_query() and do INSERT/REPLACE/DELETE as usual
- Searching is transparent
 - SphinxAPI / SphinxSE / SphinxQL all work
 - We now prefer SELECT that we have SphinxQL :)
- Some features are not yet (!) supported
 - MVA, geosearch, prefix and infix indexing support to be implemented

Beyond the basics

- Tokenizing settings
- Wordforms support
- 1-grams
- SQL and IO throttling
- Arbitrary expressions
- Prefix/infix indexing
- Blended characters
- Hitless indexing
- Several bundled rankers
- ...

More about Sphinx



Available in retail from 4/19/2011

Upcoming 2.0 release

- SphinxQL improvements
 - multi-query support
 - more SphinxQL functions and operators
- "keywords" dictionary
 - improves substring indexing a lot
- Zones, sentences, paragraphs support
- Multi-threaded snippet batches support
- UDF support (CREATE/DROP FUNCTION)
- Extended support for strings
 - ORDER BY, GROUP BY, WITHING GROUP ORDER BY
- 35+ more new features

Looking to work on Sphinx ?

Consultants, support engineers,
Q/A engineer and technical writer wanted!

<http://sphinxsearch.com/about/careers/>

Please mail us at job2011@sphinxsearch.com

Percona Live, May 26, New York



PERCONA
LIVE

www.percona.com/live