



**Full-Text Search with Sphinx and MySQL**

**DevConf, Moscow, 2011**

# What is Sphinx

---

- Free open source search server
- Begins 10 years ago as a full text daemon
- Now powerful, fast, relevant, scalable search engine.
- Dual licensing model, just like MySQL
- Available for Linux, Windows, Mac OS
  - Can be built on AIX, iPhone and some DSL routers



# What Sphinx Can Do For You?

---

- Serve 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/>

# Powerful FT-query syntax

---

- 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

# Not only Full-Text search

---

- Geo distance search
- MVA (i.e. page tags or multiple categories)
- UNIX timestamps
- Floating point values
- Strings & Integers
- Built-in expressions, functions, and operators
- UDF support

# Full-Text search with MyISAM

---

- Working out of the box
- Not always fast
  - Very slow on large datasets
- Can't scale well
- General MyISAM downsides

# Full-Text search with MyISAM

---

```
mysql> SELECT id, title FROM posting_m1
-> WHERE MATCH(title, content)
-> AGAINST ('I Love Sphinx')
-> LIMIT 10;
```

```
+-----+-----+
| id      | title                                     |
+-----+-----+
| 7387035 | |
| 6941386 | [LOTR] Eloquent Remembrance by Sphinx |
| 5926102 | I've Done it Again                     |
| 6814871 | Oh Geesh.                               |
...
| 5884330 | England, Egypt, Italy, France...      |
+-----+-----+
```

```
10 rows in set (0.11 sec)
```



# Full-Text search with Sphinx

---

```
mysql> SELECT * FROM lj1m
      -> WHERE MATCH('I Love Sphinx')
      -> LIMIT 10;
```

id	weight	channel_id	ts
7637682	2652	358842	1112905663
6598265	2612	454928	1102858275
...			
7139955	1616	403287	1080666627
5068690	1612	554732	1074928240

10 rows in set (0.00 sec)



# 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

# Query Sphinx via mysql client

---

```
$ mysql -h 0 -P 9306
Welcome to the MySQL monitor.  Commands end with ; or \g.
Your MySQL connection id is 1
Server version: 2.0.2-id64-dev (r2824)

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

```
mysql> SELECT * FROM lj 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)
```



# Integration with Sphinx

---

- SphinxQL
- API
- SphinxSE



# Sphinx API

---

```
<?php
require ( "sphinxapi.php" ); //from sphinx distro
...
$cl = new SphinxClient();
...

$res = $cl->Query ( "my first query", "my_index" );
var_dump ( $res );

?>
```

# Sphinx APIs

---

- Officially available
  - PHP
  - Python
  - Java
  - Ruby
  - Pure C
- .NET, Thinking Sphinx (for Rails) and few more available as third party plugins

# How to build a rocket?

---










# Download and Install

- <http://sphinxsearch.com/downloads/>

## Sphinx 2.0.1-beta downloads

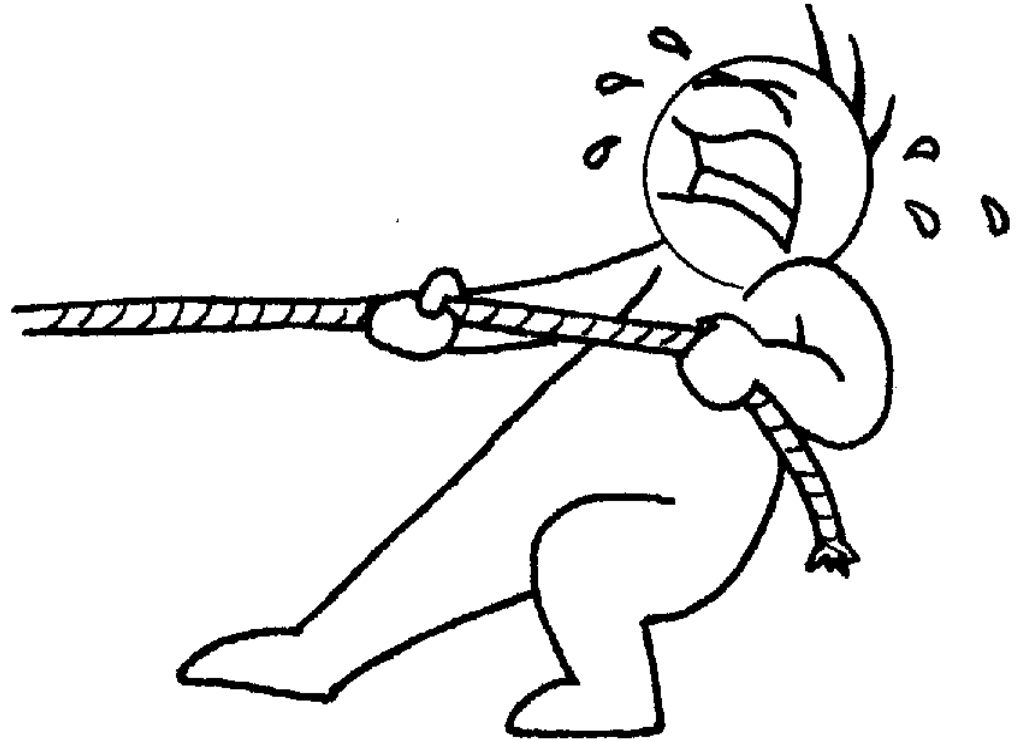
**Sphinx 2.0.1-beta** (r2792; Apr 22, 2011)

 Source tarball (tar.gz)	2.0.1-beta	1.7M
 Win32 binaries w/MySQL support	2.0.1-beta	3.8M
 Win32 binaries w/MySQL+PostgreSQL support	2.0.1-beta	5.3M
 Win32 binaries w/MySQL+PgSQL+libstemmer+id64 support	2.0.1-beta	5.6M
 RHEL/CentOS 5.x x86_64 RPM	2.0.1-beta	4.2M
 RHEL/CentOS 5.x i386 RPM	2.0.1-beta	4.8M
 Mac OS X 10.6.x i386 binaries	2.0.1-beta	10.2M

# Where to get data?

---

- MySQL
- PostgreSQL
- MSSQL
- ODBC source
- XML pipe





# MySQL source

---

```
source lj_source
{
  ...
  sql_query = \
    SELECT id, channel_id, ts, title, content \
    FROM ljposts

  sql_attr_uint      = channel_id
  sql_attr_timestamp = ts
  ...
}
```

# A complete version

---

```
source lj_source
{
    type      = mysql
    sql_host  = localhost
    sql_user  = my_user
    sql_pass  = my*****
    sql_db    = test

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

    sql_attr_uint      = channel_id
    sql_attr_timestamp = ts

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

# How to process. Index config.

---

```
index lj
{
  source          = lj_source
  path            = /my/index/path/lj_index

  html_strip     = 1
  html_index_attrs = img=src,alt; a=href,title

  morphology      = stem_en
  stopwords       = stopwords.txt
  charset_type    = utf-8
}
```

# Indexer configuration

---

```
indexer
```

```
{
```

```
    mem_limit    = 512M
```

```
    max_iops     = 40
```

```
    max_iosize  = 1048576
```

```
}
```



# Building index

---

```
$ ./indexer lj
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 'lj'...
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
```



# Index files

---

```
$ ls -lah lj*
```

```
-rw-r--r-- 1 vlad vlad 12M 2010-12-22 09:01 lj.spa  
-rw-r--r-- 1 vlad vlad 334M 2010-12-22 09:01 lj.spd  
-rw-r--r-- 1 vlad vlad 438 2010-12-22 09:01 lj.sph  
-rw-r--r-- 1 vlad vlad 13M 2010-12-22 09:01 lj.spi  
-rw-r--r-- 1 vlad vlad 0 2010-12-22 09:01 lj.spk  
-rw-r--r-- 1 vlad vlad 0 2011-05-13 09:25 lj.spl  
-rw-r--r-- 1 vlad vlad 0 2010-12-22 09:01 lj.spm  
-rw-r--r-- 1 vlad vlad 111M 2010-12-22 09:01 lj.spp  
-rw-r--r-- 1 vlad vlad 1 2010-12-22 09:01 lj.sps
```

```
$
```

# Configuring searchd

---

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

    preopen_indexes      = 1
    max_packet_size      = 8M

    query_log_format     = sphinxql
    query_log             = query.log

    pid_file              = searchd.pid
}
```





# Starting sphinx!

---

```
$ ../bin/searchd -c sphinx.conf
Sphinx 2.0.2-dev (r2824)
Copyright (c) 2001-2010, Andrew Aksyonoff
Copyright (c) 2008-2010, Sphinx Technologies
  Inc (http://sphinxsearch.com)

using config file 'sphinx.conf'...
listening on 127.0.0.1:9312
listening on 127.0.0.1:9306
precaching index 'lj'
precached 1 indexes in 0.028 sec
```



# Search with SphinxQL

```
mysql> SELECT *  
      -> FROM lj1m  
      -> 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)
```

# Faceted search

```
mysql> SELECT *, YEAR(ts) 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);
```

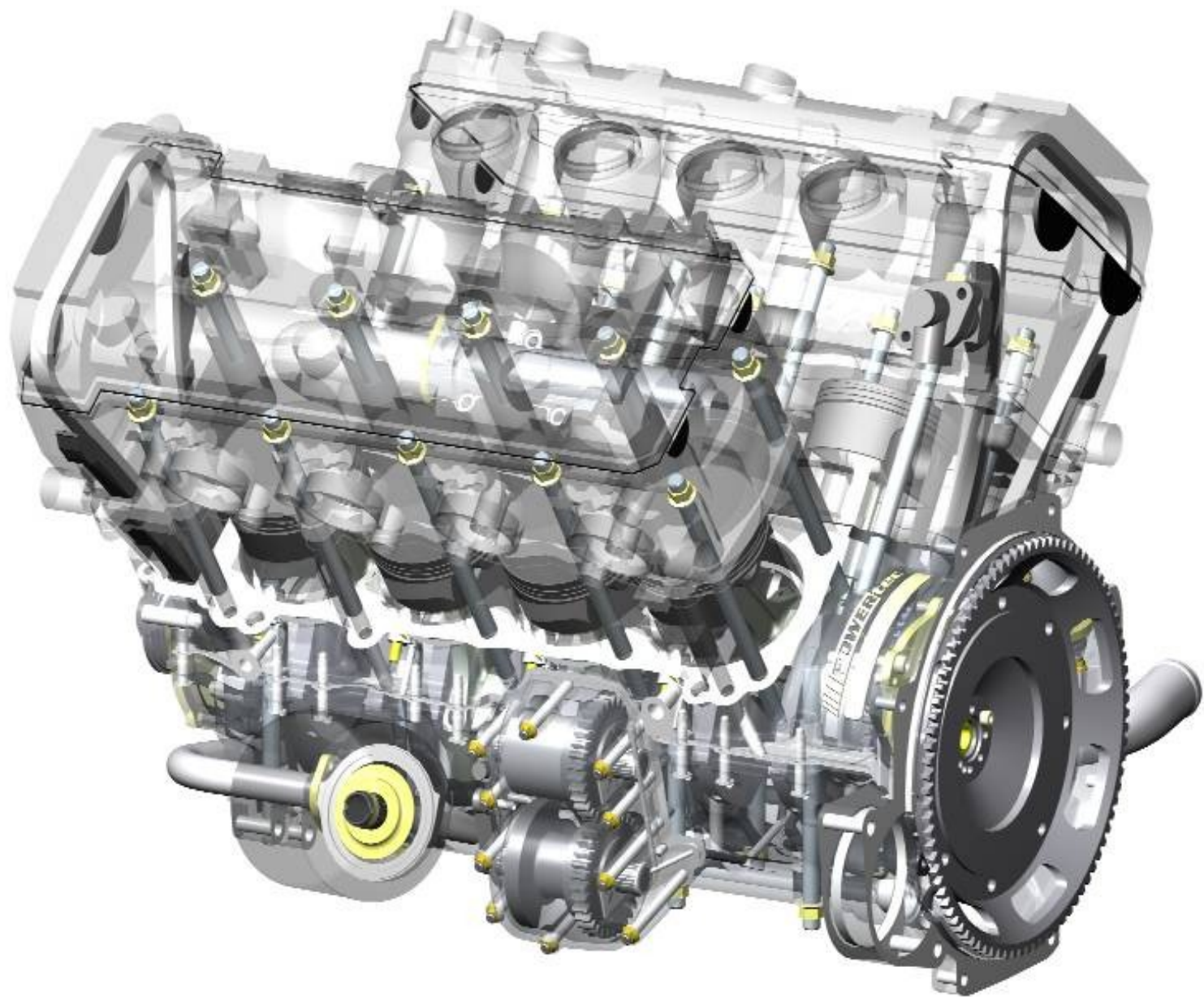
id	weight	channel_id	ts	yr	@groupby	@count
7637682	101652	358842	1112905663	2005	2005	14
6598265	101612	454928	1102858275	2004	2004	27
7139960	1642	403287	1070220903	2003	2003	8
5340114	1612	537694	1020213442	2002	2002	1
5744405	1588	507895	995415111	2001	2001	1

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

# Typical Sphinx applications

---

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



# 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
  - group\_concat emulation for non MySQL sources
  - 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 | 1,4      |  
+-----+-----+-----+  
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)
```



# Geodistance 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 (0.05 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, pgsql, mssql sources only
- Since 1.10-beta

# RT indexes

---

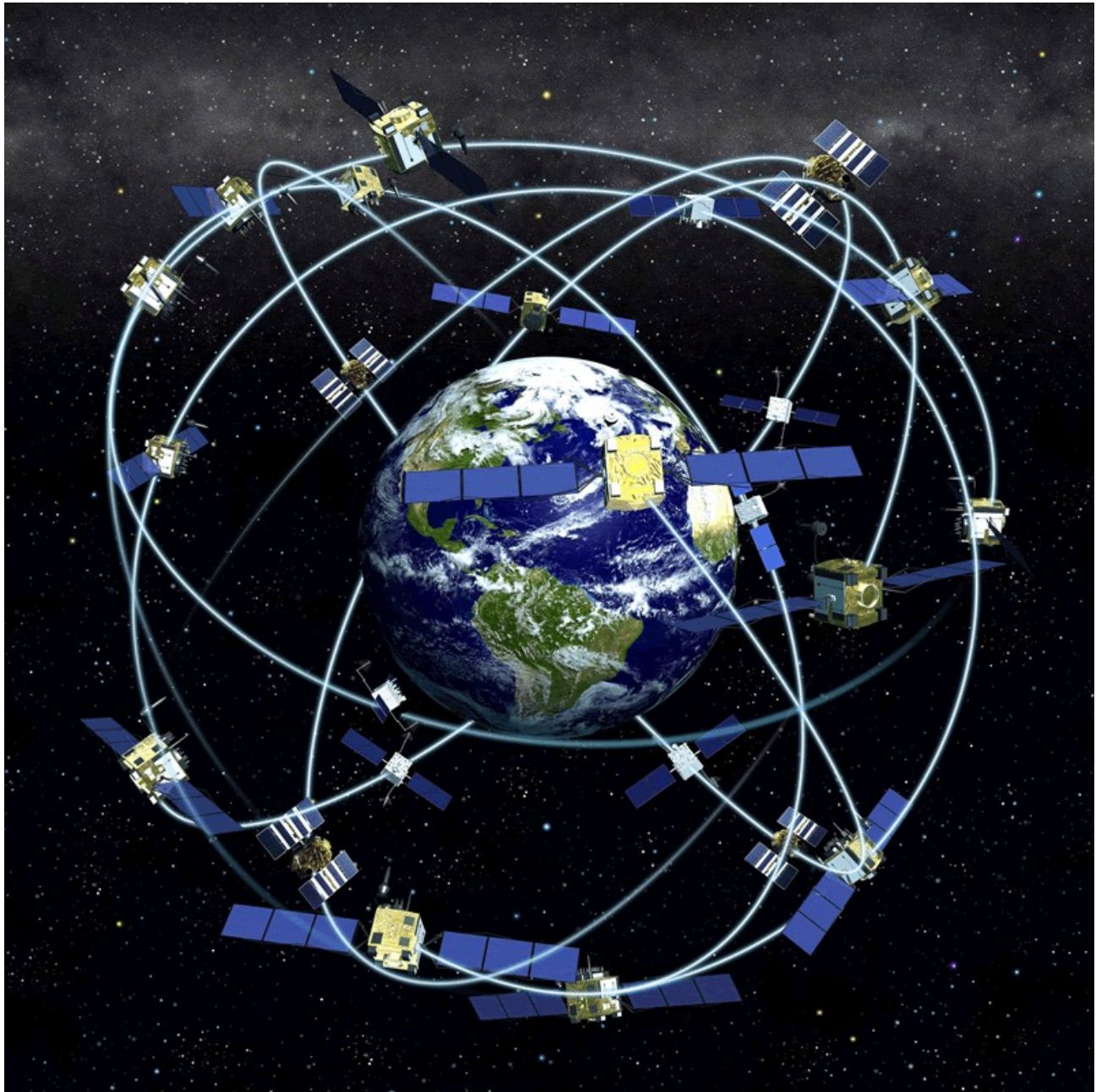
- Push model instead of Pull for on-disk indexes
  - via INSERT/UPDATE/DELETE
- Update data on the fly
- Formally “soft-realtime”
  - As in, most of the writes are very quick
  - But, not guaranteed to complete in fixed time
- Transparent for application

# 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





# Scale!

---

- Working out of the box
- Shard the data
  - Within a box to utilize multicore servers
  - Spread load across several boxes

# Part two: local indexes

---

```
index my_distributed_index1
{
    type          = distributed
    local         = ondisk_index1
    local         = ondisk_index2
    local         = ondisk_index3
    local         = ondisk_index4
}
...
dist_threads = 4
...
```

# Part three: distributed indexes

---

```
index my_distribited_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
}
```

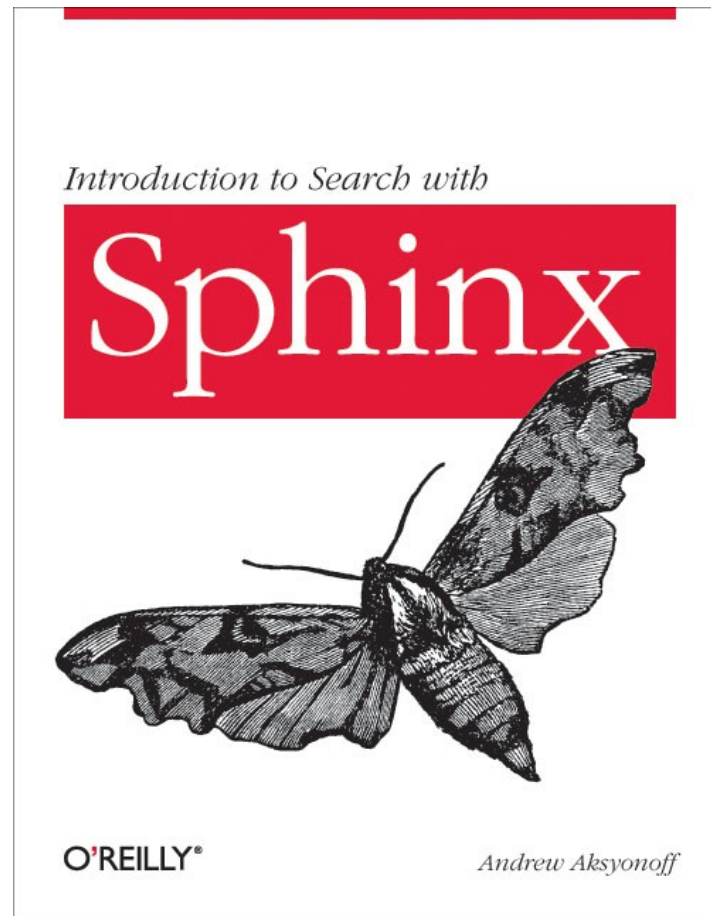
# 2.0 release

---

- 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

# More about Sphinx

---



# Sphinx today

---

## We're hiring!

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

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

Just let me know or  
mail us at [job2011@sphinxsearch.com](mailto:job2011@sphinxsearch.com)



Questions?



<http://sphinxsearch.com>