Full Text Search in MySQL 5.1 New Features and HowTo

Alexander Rubin

Senior Consultant, MySQL AB

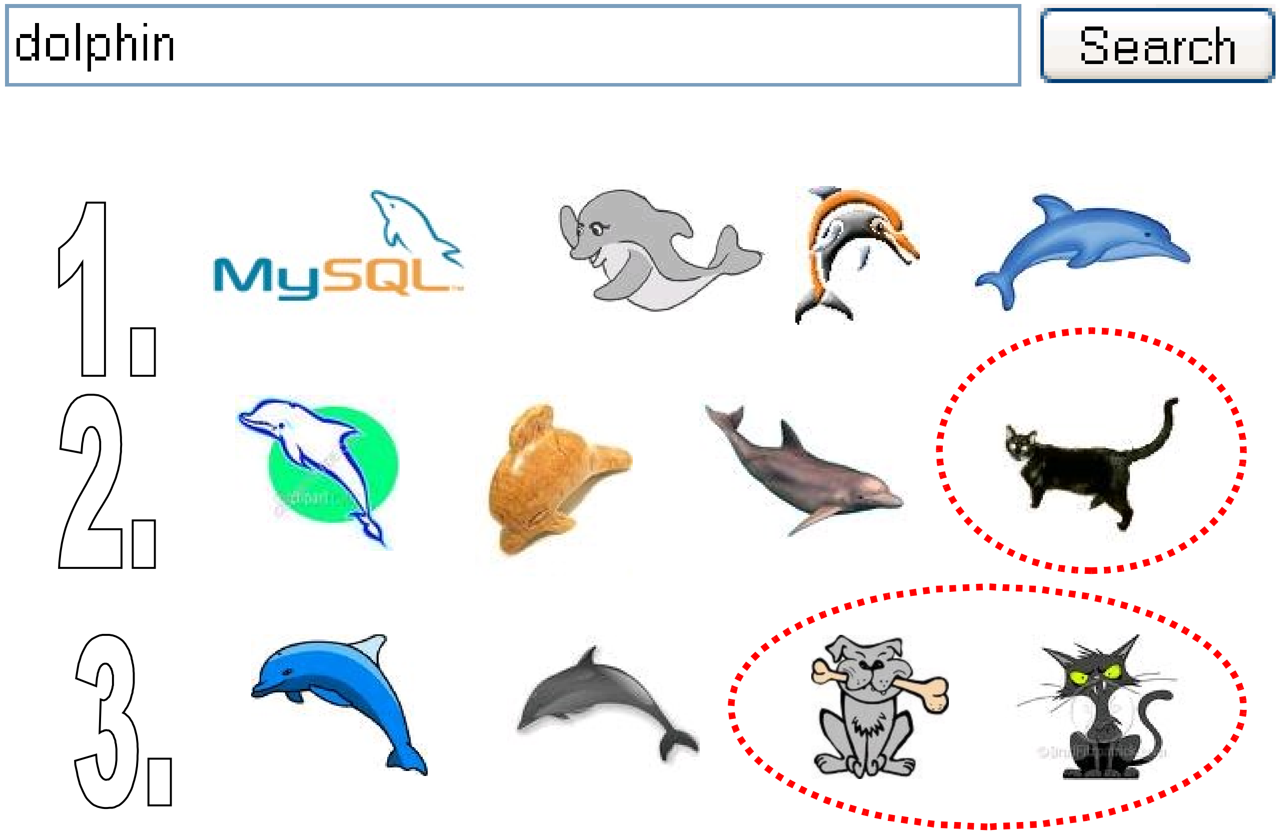
Full Text search

- Natural and popular way to search for information
- Easy to use: enter key words and get what you need

In this presentation

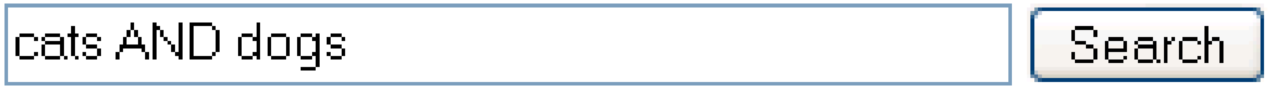
- Improvements in FT Search in MySQL 5.1
- How to speed up MySQL FT Search
- How to search with error corrections
- Benchmark results

Types of FT Search: Relevance

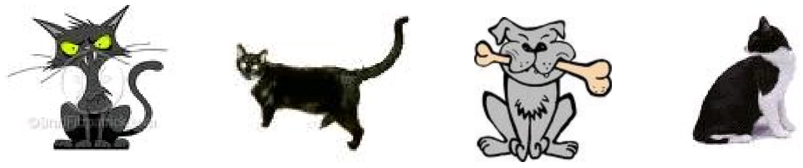


- MySQL FT: Default by relevance!

Types of FT Search: Boolean Search


cats AND dogs

1.

2.


3.

- MySQL FT: No default sorting!

Types of FT Search: Phrase Search

"dolphin in the sun"

Search Images

Full Text Solutions

Type	***Solution***
MySQL Built-in	Full Text Index (MyISAM only)
MySQL Integrated/External	Sphinx
External	Lucene MnogoSearch
“Hardware boxes”	Google box “Fast” box

MySQL Full Text Index Features

- Available only for MyISAM tables
- Natural Language Search and boolean search
- ft_min_word_len – 4 char per word by default
- Stop word list by default
- Frequency based ranking
  - Distance between words is not counted

MySQL Full Text: Creating Full Text Index

```
mysql> CREATE TABLE articles (
-> id INT UNSIGNED
  AUTO_INCREMENT NOT NULL
  PRIMARY KEY,
-> title VARCHAR(200),
-> body TEXT,
-> FULLTEXT (title,body)
-> ) engine=MyISAM;
```

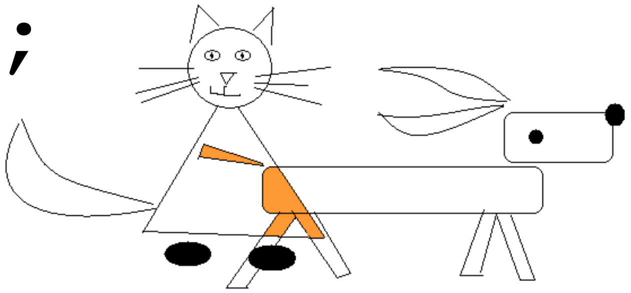
MySQL Full Text: Natural Language mode

```
mysql> SELECT * FROM articles
-> WHERE MATCH (title,body)
-> AGAINST ('database' IN
  NATURAL LANGUAGE MODE);
+-------------------+------------------------------------+
| title | body                                           |
+-------------------+------------------------------------+
| MySQL vs. YourSQL | In the following database comparison ... |
| MySQL Tutorial | DBMS stands for DataBase ...            |
```

In Natural Language Mode:
default sorting by relevance!

MySQL Full Text: Boolean mode

```
mysql> SELECT * FROM
  articles
-> WHERE MATCH (title,body)
-> AGAINST ('cat AND dog'
  IN BOOLEAN MODE);
```



No default sorting in Boolean Mode!

New MySQL 5.1 Full Text Features

New MySQL 5.1 Full Text Features

- ***Faster Boolean search in MySQL 5.1***
  - New “smart” index merge is implemented

  (forge.mysql.com/worklog/task.php?id=2535)

- ***Custom Plug-ins***
  - Replacing Default Parser

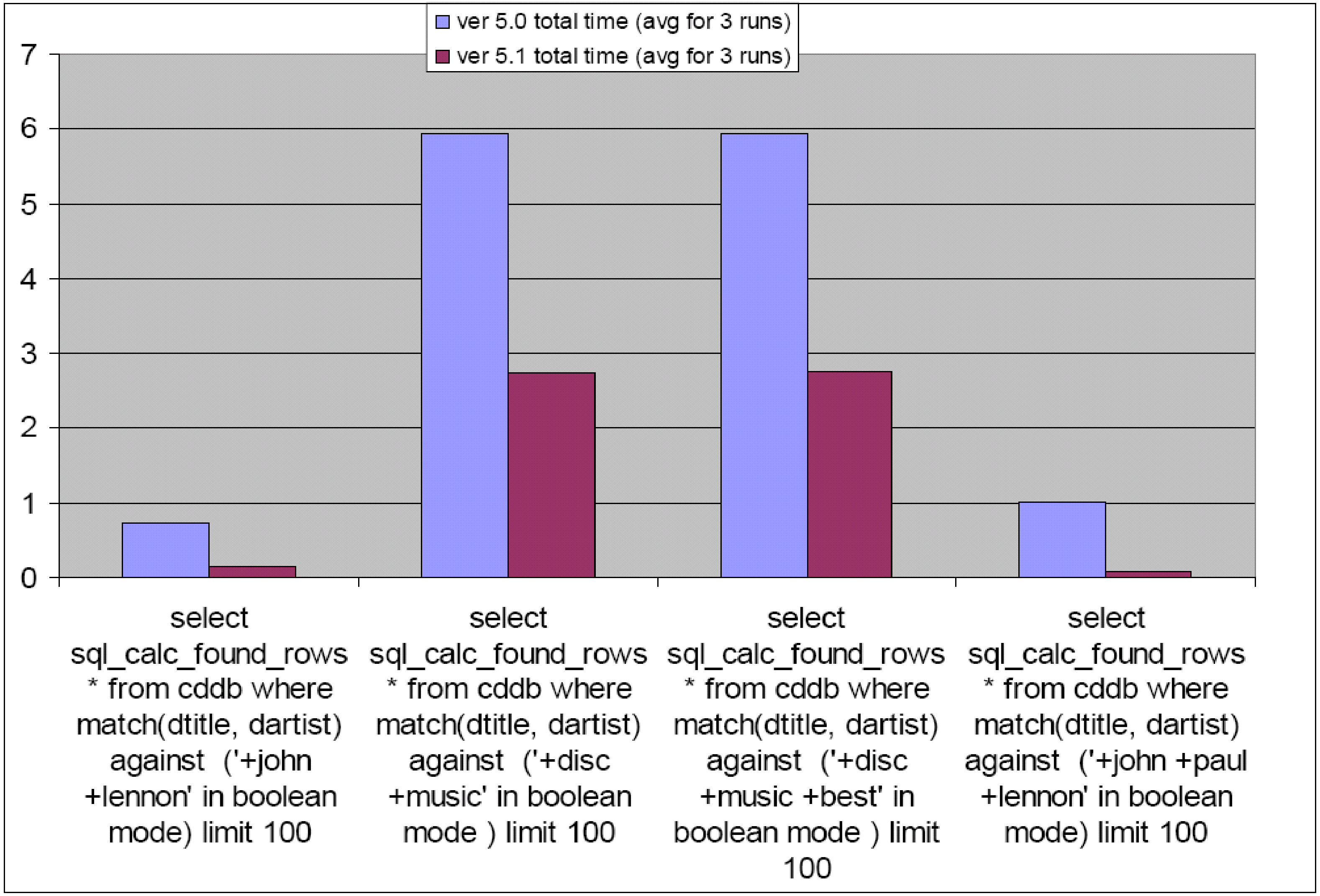
- ***Better Unicode Support***
  - full text index work more accurately with space and punctuation Unicode character

  (***forge.mysql.com/worklog/task.php?id=1386***)

MySQL 5.0 vs 5.1 Benchmark

- MySQL 5.1 Full Text search: 500-1000% improvement in Boolean mode
  - relevance based search and phrase search was not improved in MySQL 5.1.
- Tested with: Data and Index
  - CDDB (music database)
  - author and title, 2 mil. CDs., varchar(255).
  - CDDB ~= amazon.com's CD/books inventory

MySQL 5.1: 500-1000% improvement in Boolean mode



MySQL 5.1 Full Text: Custom “Plugins”

- Replacing Default Parser to do following:
  - Apply special rules, such as stemming, different way of splitting words etc
  - Pre-parsing – processing PDF / HTML files

- May do same for query string
  - If use stemming, search words also need to be stemmed for search to work.

Available Plugins Examples

- Different plugins: search for “FullText” at forge.mysql.com/projects
- Stemming
  - MnogoSearch has a stemming plugin (www.mnogosearch.com)
  - Porter stemming fulltext plugin
- N-Gram parsers (Japanese language)
  - Simple n-gram fulltext plugin

Example: MnogoSearch Stemming

- MnogoSearch includes stemming plugin (www.mnogosearch.org/doc/msearch-udmstemmer.html)
- Configure and install:
  – Configure (follow instructions)
  – mysql> INSTALL PLUGIN stemming SONAME 'libmnogosearch.so';
  – CREATE TABLE my_table (my_column TEXT, FULLTEXT(my_column) ***WITH PARSER stemming***);
  – SELECT * FROM t WHERE MATCH a AGAINST('test' IN BOOLEAN MODE);

Example: MnogoSearch Stemming

- Configuration: stemming.conf
  - MinWordLength 2
  - Spell en latin1 american.xlg
  - Affix en latin1 english.aff
- Grab Ispell (not Aspell) dictionaries from http://lasr.cs.ucla.edu/geoff/ispell-dictionaries.html#English-dicts
- Any changes in stemming.conf requires MySQL restart

Example: MnogoSearch Stemming

Stemming adds overhead on insert/update

```
mysql> insert into searchindex_stemmer
select * from enwiki.searchindex limit
10000;
Query OK, 10000 rows affected (44.03
sec)


mysql> insert into searchindex select
* from enwiki.searchindex limit 10000;
Query OK, 10000 rows affected (21.80
sec)
```

Example: MnogoSearch Stemming

```
mysql>  SELECT count(*) FROM
searchindex WHERE MATCH si_text
AGAINST('color' IN BOOLEAN
MODE);
```

```
count(*): 861
```

```
mysql>  SELECT count(*) FROM
searchindex_stemmer WHERE MATCH
si_text AGAINST('color' IN
BOOLEAN MODE);
```

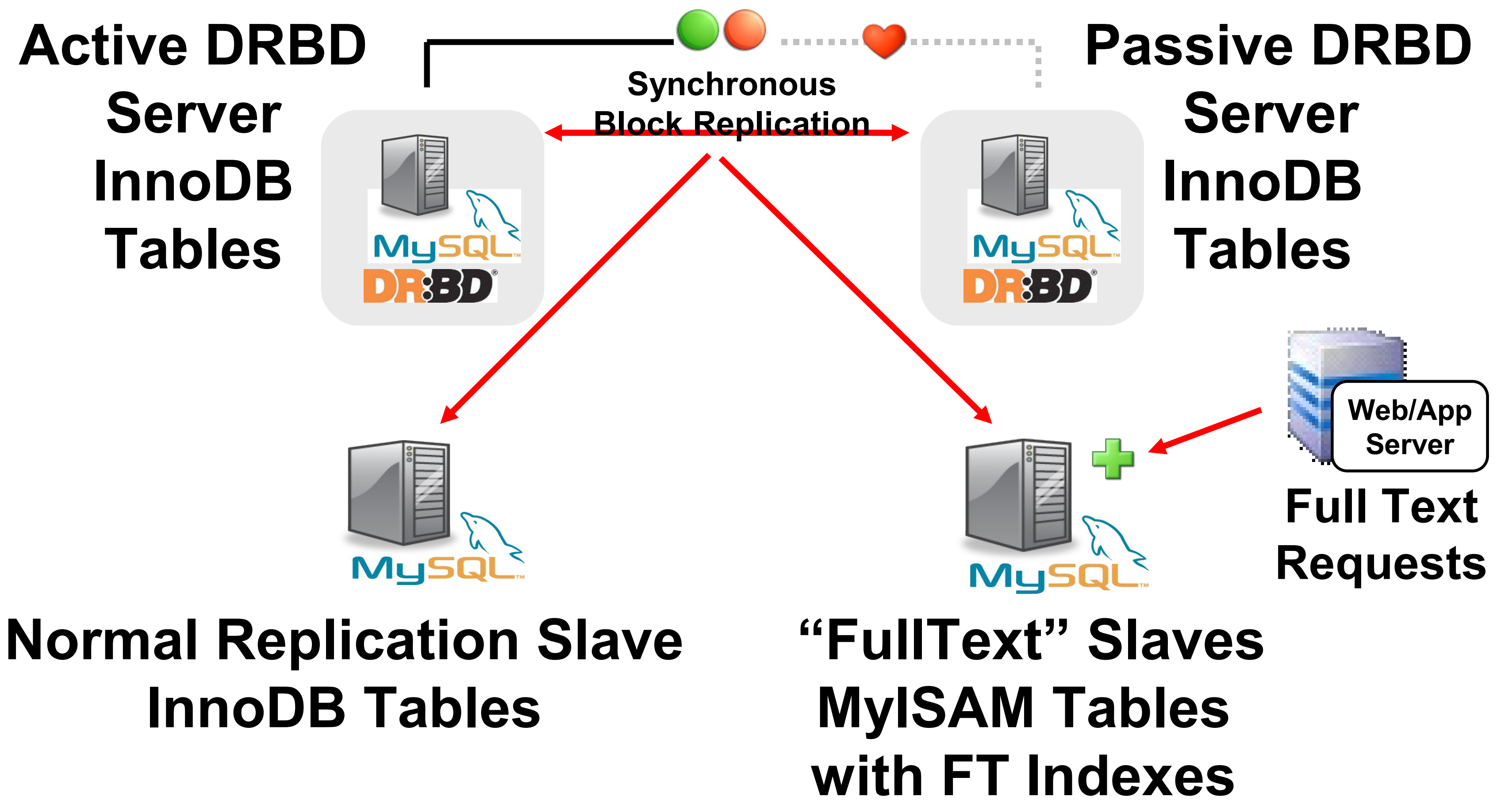
```
count(*): 1017
```

Other Planned Full Text Features

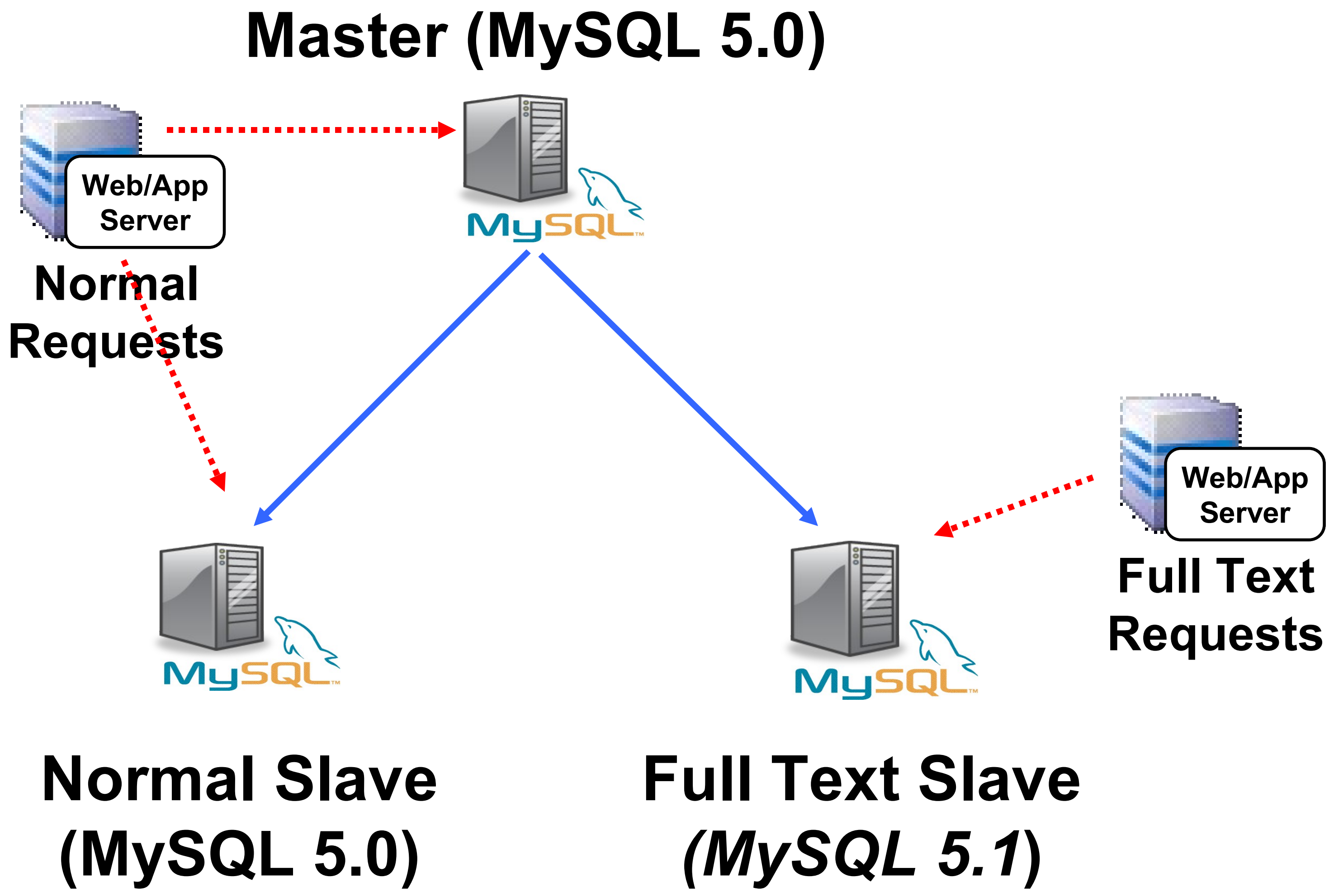
- Search for “FullText” at forge.mysql.com
- CTYPE table for unicode character sets (WL#1386), complete
- Enable fulltext search for non-MyISAM engines (WL#2559), Assigned
- Stemming for fulltext (WL#2423), Assigned
- Combined BTREE/FULLTEXT indexes (WL#828)
- Many other features, YOU can vote for some

MySQL Full Text HowTo Tips and Tricks

DRBD and FullText Search


Active DRBD Server InnoDB Tables
Synchronous Block Replication
Passive DRBD Server InnoDB Tables
Web/App Server
Full Text Requests
Normal Replication Slave InnoDB Tables
"FullText" Slaves MyISAM Tables with FT Indexes

Using MySQL 5.1 as a Slave



How To: Speed up MySQL FT Search

Fit index into memory!

- Increase amount of RAM
- Set key_buffer = <total size of full text index>. *Max 4GB!*
- Preload FT indexes into buffer
  - Use additional keys for FT index (to solve 4GB limit problem)

SpeedUp FullText: Preload FT indexes into buffer

```
mysql>  set global
 ft_key.key_buffer_size=
4*1024*1024*1024;
mysql>  CACHE INDEX S1, S2,
 <some other tables here>
 IN ft_key;
mysql>  LOAD INDEX INTO
 CACHE S1, S2 …;
```

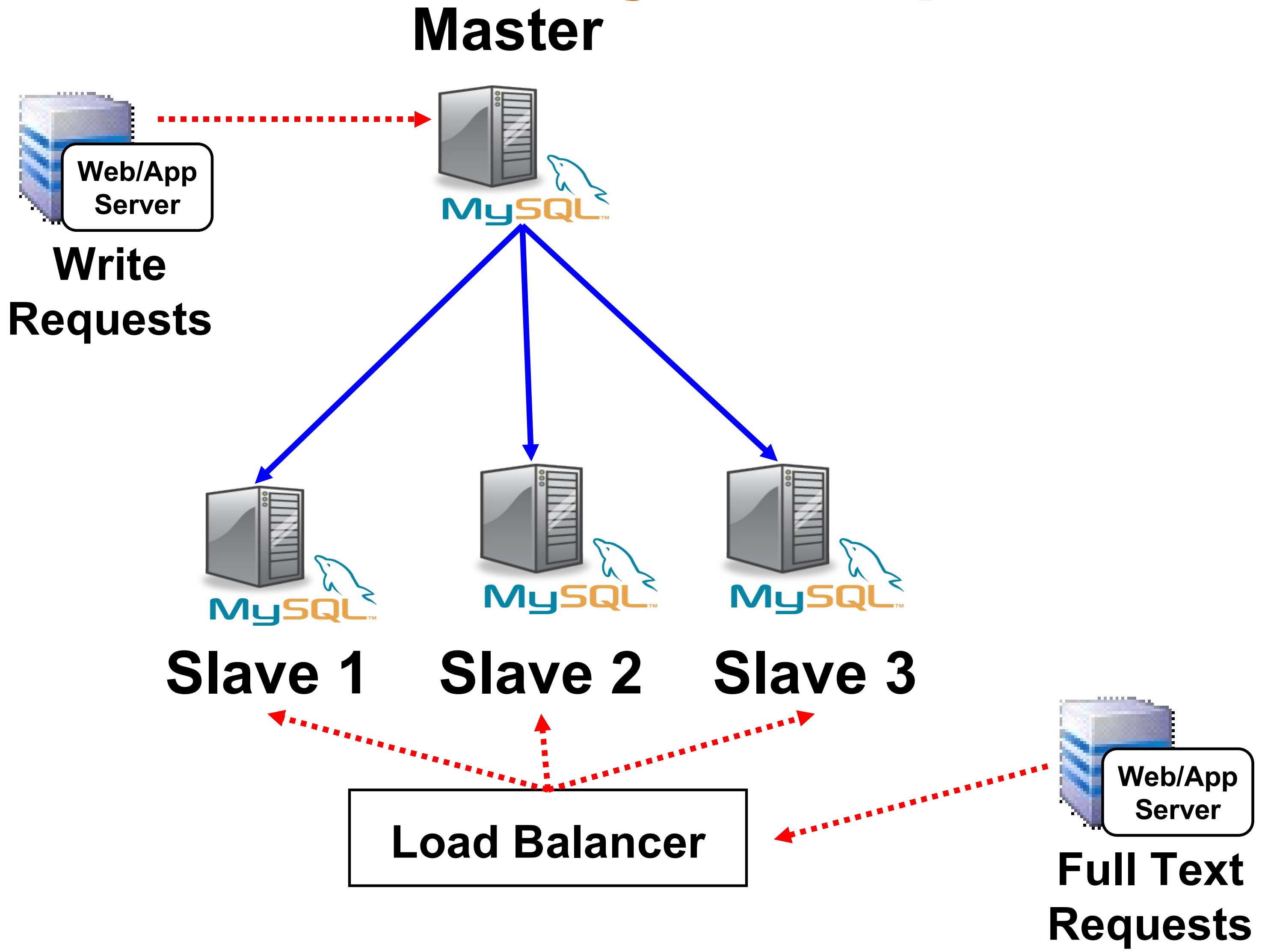
How To: Speed up MySQL FT Search

- Manual partitioning
  - Partitioning will decrease index and table size
    - Search and updates will be faster
  - Need to change application/no auto partitioning

How To: Speed up MySQL FT Search

- Setup number of slaves for search
  - Decrease number of queries for each box
  - Decrease CPU usage (sorting is expensive)
  - Each slave can have its own data
    - Example: search for east coast – Slaves 1-5, search for west coast Slaves 6-10

FT Scale-Out with MySQL Replication



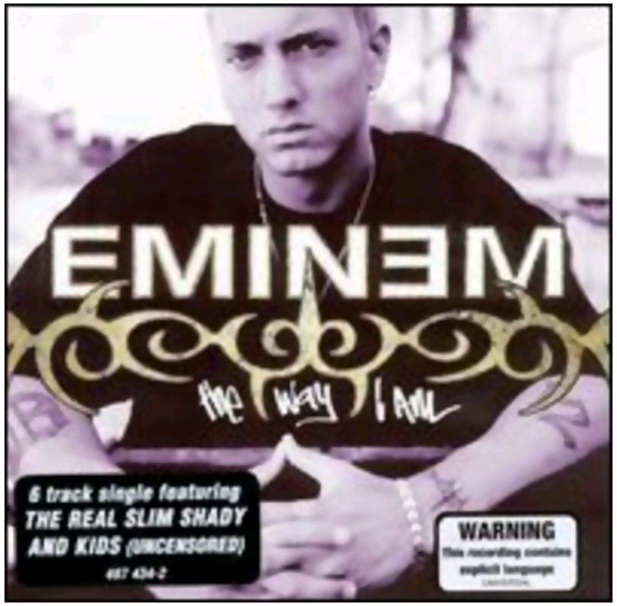
Which queries are performance killers

- Order by/Group by
  - Natural language mode: order by relevance
  - Boolean mode: no default sorting!
  - ***Order by date much slower than with no "order by"***
- SQL_CALC_FOUND_ROWS
  - Select SQL_CALC_FOUND_ROWS from T1 … limit 10
  - Will require all result set
- Other condition in where clause
  - **MySQL can use either FT index or other indexes (only FT with natural language)**

Real World Example: Performance Killer

Why it is so slow?

```
SELECT … FROM `ft`
WHERE MATCH `album`
AGAINST
('the way i am')
```



Real World Example: Performance Killer

Note the stopword list and
ft_min_word_len!

~~The~~	–	*stopword*
~~Way~~	–	*stopword*
I	–	not a stop word
~~Am~~	–	*stopword*

My.cnf:

ft_min_word_len =1

query *"the way i am"* will filter out all words except *"i"* with standard stoplist and with ft_min_word_len =1 in my.cnf

How To: Search with error correction

- Example: Music Search Engine
  - Search for music titles/actors
  - Need to correct users typos
    - ***Bob Dilane*** (user made typos) ->
      - **Bob Dylan** (corrected)
- Solution:
  - use soundex() mysql function
    - Soundex = sounds similar

```
mysql> select soundex('Dilane');
D450
mysql> select soundex('Dylan');
D450
```

Search with error corrections

- Implementation

```
1.Alter table artists add
  art_name_sndex varchar(80)
2.Update artists set
  art_name_sndex =
  soundex(art_name)
3.Select art_name from artists
  where art_name_sndex =
  soundex(‘Bob Dilane') limit 10
```

Search with error corrections: Sorting

- Popularity of the artist
  - `Select art_name from artists where art_name_sndex = soundex('Dilane') order by popularity limit 10`
- Most similar matches fist – order by levenstein distance
  - The Levenshtein distance between two strings = minimum number of operations needed to transform one string into the other
  - Levenstein can be done by stored function or UDF

Sphinx Search

- Features
  - Open Source, http://www.sphinxsearch.com
  - Fast searches for the large data
  - Designed for indexing Database content
  - Supports multi-node clustering out of box, Multiple Attributes (date, price, etc)
  - Different sort modes (relevance, data etc)
  - Client available as MySQL Storage Engine plugin
  - Fast Index creation (up to 10M/sec)
- Disadvantages:
  - External solution: need to be integrated
  - No online index updates (have to build whole index)

How to configure Sphinx with MySQL

- Sphinx engine/plugin is not full engine:
  - still need to run “searcher” daemon
- Need to compile MySQL source with Sphinx to integrate it
  - MySQL 5.0: need to patch source code
  - MySQL 5.1: no need to patch, copy Sphinx plugin to plugin dir

How to Integrate Sphinx with MySQL

- ***Sphinx can be MySQL's storage engine***

```
CREATE TABLE t1
    (id               INTEGER NOT NULL,
     weight           INTEGER NOT NULL,
     query            VARCHAR(3072) NOT NULL,
     group_id         INTEGER,
     INDEX(query)
    ) ENGINE=SPHINX
     CONNECTION="sphinx://localhost:
     3312/enwiki";

   SELECT * FROM enwiki.searchindex docs
   JOIN test.t1 ON (docs.si_page=t1.id)
   WHERE query="one document;mode=any"
     limit 1;
```

Time for questions

Questions?

www.mysqlfulltextsearch.com