

二

32 实战：Redisearch 高性能的全文搜索引擎

Redisearch 是一个高性能的全文搜索引擎，它可以作为一个 Redis Module（扩展模块）运行在 Redis 服务器上。

Redisearch 主要特性如下：

- 基于文档的多个字段全文索引
- 高性能增量索引
- 文档排序（由用户在索引时手动提供）
- 在子查询之间使用 AND 或 NOT 操作符的复杂布尔查询
- 可选的查询子句
- 基于前缀的搜索
- 支持字段权重设置
- 自动完成建议（带有模糊前缀建议）
- 精确的短语搜索
- 在许多语言中基于词干分析的查询扩展
- 支持用于查询扩展和评分的自定义函数
- 将搜索限制到特定的文档字段
- 数字过滤器和范围
- 使用 Redis 自己的地理命令进行地理过滤
- Unicode 支持（需要 UTF-8 字符集）
- 检索完整的文档内容或只是 ID 的检索
- 支持文档删除和更新与索引垃圾收集
- 支持部分更新和条件文档更新

安装

和前面讲到布隆过滤器的引入方式一样, 我们可以使用 RedisSearch 官方推荐的 Docker 方式来安装并启动 RedisSearch 功能, 操作命令如下:

```
docker run -p 6379:6379 redislabs/redisearch:latest
```

安装并启动成功, 如下图所示:

```
latest: Pulling from redislabs/redisearch
1ab2bdf9e9778: Pull complete
e3286b9257d7: Pull complete
413ee3ee3fa2: Pull complete
47c871a613d4: Pull complete
7198bcb813b4: Pull complete
e57e6f7405b1: Pull complete
412212364483: Pull complete
651c0a9e30b7: Pull complete
84e26e50ba7f: Pull complete
3a1d6f64ec79: Pull complete
Digest: sha256:a1e1b2407919e669f7e986dc254f61fa32897c02b54464413812227895e7381d
Status: Downloaded newer image for redislabs/redisearch:latest
1:C 21 Mar 2020 13:12:44.640 # o000o000o000o Redis is starting o000o000o000o
1:C 21 Mar 2020 13:12:44.640 # Redis version=5.0.7, bits=64, commit=00000000, modified=0, pid=1, just started
1:C 21 Mar 2020 13:12:44.640 # Configuration loaded
1:M 21 Mar 2020 13:12:44.641 * Running mode=standalone, port=6379.
1:M 21 Mar 2020 13:12:44.641 # WARNING: The TCP backlog setting of 511 cannot be enforced because /proc/sys/net/core/somaxconn is set to the lower value of 128.
1:M 21 Mar 2020 13:12:44.641 # Server initialized
1:M 21 Mar 2020 13:12:44.641 # WARNING you have Transparent Huge Pages (THP) support enabled in your kernel. This will create latency and memory usage issues with Redis. To fix this issue run the command 'echo never > /sys/kernel/mm/transparent_hugepage/enabled' as root, and add it to your /etc/rc.local in order to retain the setting after a reboot. Redis must be restarted after THP is disabled.
1:M 21 Mar 2020 13:12:44.641 * <ft> RedisSearch version 1.6.10 (Git=v1.6.10)
1:M 21 Mar 2020 13:12:44.641 * <ft> Low level api version 1 initialized successfully
1:M 21 Mar 2020 13:12:44.641 * <ft> concurrent writes: OFF, gc: ON, prefix min length: 2, prefix max expansions: 200, query timeout (ms): 500, timeout policy: return, cursor read size: 1000, cursor max idle (ms): 300000, max doc table size: 10000000, search pool size: 20, index pool size: 8,
1:M 21 Mar 2020 13:12:44.641 * <ft> Initialized thread pool!
1:M 21 Mar 2020 13:12:44.641 * Module 'ft' loaded from /usr/lib/redis/modules/redisearch.so
1:M 21 Mar 2020 13:12:44.642 * Ready to accept connections
```

安装完成之后使用 **redis-cli** 来检查 RedisSearch 模块是否加载成功, 使用 Docker 启动 redis-cli, 命令如下:

```
docker exec -it myredis redis-cli
```

其中“myredis”为 Redis 服务器的名称, 执行结果如下:

```
127.0.0.1:6379> module list
1) 1) "name"
   2) "ft"
   3) "ver"
   4) (integer) 10610
```

返回数组存在“ft”，表明 Redisearch 模块已经成功加载。

源码方式安装

如果不想使用 Docker，我们也可以使用源码的方式进行安装，安装命令如下：

```
git clone https://github.com/RedisLabsModules/Redisearch.git
cd Redisearch # 进入模块目录
make all
```

安装完成之后，可以使用如下命令启动 Redis 并加载 Redisearch 模块，命令如下：

```
src/redis-server redis.conf --loadmodule ../Redisearch/src/redisearch.so
```

使用

我们先使用 redis-cli 来对 Redisearch 进行相关的操作。

创建索引和字段

```
127.0.0.1:6379> ft.create myidx schema title text weight 5.0 desc text
OK
```

其中“myidx”为索引的ID，此索引包含了两个字段“title”和“desc”，“weight”为权重，默认值为1.0。

将内容添加到索引

```
127.0.0.1:6379> ft.add myidx doc1 1.0 fields title "He urged her to study English"
OK
```

其中“doc1”为文档 ID（docid），“1.0”为评分（score）。

根据关键查询

```
127.0.0.1:6379> ft.search myidx "english" limit 0 10
1) (integer) 1
2) "doc1"
3) 1) "title"
   2) "He urged her to study English"
   3) "desc"
   4) "good idea"
```

可以看出我们使用 title 字段中的关键字“english”查询出了一条满足查询条件的数据。

中文搜索

首先我们需要先给索引中，添加一条中文数据，执行命令如下：

```
127.0.0.1:6379> ft.add myidx doc2 1.0 language "chinese" fields title "Java 14 发布
OK
```

注意：**这里必须要设置语言编码为中文，也就是“language "chinese"”**，默认是英文编码，如果不设置则无法支持中文查询（无法查出结果）。

我们使用之前的查询方式，命令如下：

```
127.0.0.1:6379> ft.search myidx "正式版"
1) (integer) 0
```

我们发现并没有查到任何信息，这是因为我们没有指定搜索的语言，不但保存时候要指定编码，查询时也需要指定，查询命令如下：

```
127.0.0.1:6379> ft.search myidx "发布了" language "chinese"
1) (integer) 1
2) "doc2"
3) 1) "desc"
   2) "Java 14 \xe5\x9c\xa8 2020.3.17 \xe6\x97\xa5\xe5\x8f\x91\xe5\xb8\x83\xe6\xad\
   3) "title"
   4) "Java 14 \xe5\x8f\x91\xe5\xb8\x83\xe4\xba\x86\xef\xbc\x81\xe6\x96\xb0\xe5\x8a
```

从结果可以看出中文信息已经被顺利的查询出来了。

删除索引的数据

```
127.0.0.1:6379> ft.del myidx doc1
(integer) 1
```

我们使用索引加文档 ID 就可以实现删除数据的功能。

删除索引

我们可以使用“ft.drop”关键字删除整个索引，执行命令如下：

```
127.0.0.1:6379> ft.drop myidx
OK
```

查询索引详细信息

我们可以使用“ft.info”关键查询索引相关信息，执行命令如下：

```
127.0.0.1:6379> ft.info myidx
1) index_name
2) myidx
3) index_options
4) (empty list or set)
5) fields
6) 1) 1) title
    2) type
    3) TEXT
    4) WEIGHT
    5) "5"
   2) 1) desc
    2) type
    3) TEXT
    4) WEIGHT
    5) "1"
7) num_docs
8) "2"
9) max_doc_id
10) "2"
11) num_terms
12) "9"
13) num_records
14) "18"
15) inverted_sz_mb
16) "0.000102996826171875"
17) total_inverted_index_blocks
18) "29"
19) offset_vectors_sz_mb
20) "1.71661376953125e-05"
21) doc_table_size_mb
```

```
22) "0.000164031982421875"
23) sortable_values_size_mb
24) "0"
25) key_table_size_mb
26) "8.0108642578125e-05"
27) records_per_doc_avg
28) "9"
29) bytes_per_record_avg
30) "6"
31) offsets_per_term_avg
32) "1"
33) offset_bits_per_record_avg
34) "8"
35) gc_stats
36) 1) bytes_collected
    2) "0"
    3) total_ms_run
    4) "16"
    5) total_cycles
    6) "14"
    7) avarage_cycle_time_ms
    8) "1.1428571428571428"
    9) last_run_time_ms
    10) "2"
    11) gc_numeric_trees_missed
    12) "0"
    13) gc_blocks_denied
    14) "0"
37) cursor_stats
38) 1) global_idle
    2) (integer) 0
    3) global_total
    4) (integer) 0
    5) index_capacity
    6) (integer) 128
    7) index_total
    8) (integer) 0
```

其中“num_docs”表示存储的数据数量。

代码实战

Redisearch 支持的客户端有以下这些。

Language	Library	Author	License	Comments
Python	redisearch-py	Redis Labs	BSD	Usually the most up-to-date client library
Java	JRedisearch	Redis Labs	BSD	

Java	lettusearch	Redis Labs	Apache-2.0	
Java	spring-redisearch	Redis Labs	Apache-2.0	
Go	redisearch-go	Redis Labs	BSD	
JavaScript	RedRedisearch	Kyle J. Davis	MIT	Partial API, compatible with Reds
JavaScript	redis-redisearch	Kyle J. Davis	MIT	
C#	NRedisearch	Marc Gravell	MIT	Part of StackExchange.Redis
PHP	redisearch-php	Ethan Hann	MIT	
Ruby on Rails	redi_search_rails	Dmitry Polyakovsky	MIT	
Ruby	redisearch-rb	Victor Ruiz	MIT	
Ruby	redi_search	Nick Pezza	MIT	Also works with Ruby on Rails

本文我们使用 JRedisearch 来实现全文搜索的功能，首先在 pom.xml 添加 JRedisearch 引

用：

```
<!-- https://mvnrepository.com/artifact/com.redislabs/jredisearch -->
<dependency>
  <groupId>com.redislabs</groupId>
  <artifactId>jredisearch</artifactId>
  <version>1.3.0</version>
</dependency>
```

完整的操作代码如下：

```
import io.redisearch.client.AddOptions;
import io.redisearch.client.Client;
import io.redisearch.Document;
import io.redisearch.SearchResult;
import io.redisearch.Query;
import io.redisearch.Schema;

public class RedisearchExample {
    public static void main(String[] args) {
        // 连接 Redis 服务器和指定索引
        Client client = new Client("myidx", "127.0.0.1", 6379);
```

```
// 定义索引
Schema schema = new Schema().addTextField("title",
    5.0).addTextField("desc", 1.0);
// 删除索引
client.dropIndex();
// 创建索引
client.createIndex(schema, Client.IndexOptions.Default());
// 设置中文编码
AddOptions addOptions = new AddOptions();
addOptions.setLanguage("chinese");
// 添加数据
Document document = new Document("doc1");
document.set("title", "天气预报");
document.set("desc", "今天的天气很好, 是个阳光明媚的大晴天, 有蓝蓝的天空和白白的");
// 向索引中添加文档
client.addDocument(document, addOptions);
// 查询
Query q = new Query("天气") // 设置查询条件
    .setLanguage("chinese") // 设置为中文编码
    .limit(0,5);
// 返回查询结果
SearchResult res = client.search(q);
// 输出查询结果
System.out.println(res.docs);
}
}
```

以上程序执行结果如下:

```
[{"id":"doc1","score":1.0,"properties":{"title":"天气预报","desc":"今天的天气很好, 是
```

可以看出添加的中文数据, 被正确的查询出来了。

小结

本文我们使用 Docker 和 源码编译的方式成功的启动了 Redisearch 功能, 要使用 Redisearch 的全文搜索功能, 必须先要创建一个索引, 然后再索引中添加数据, 再使用 `ft.search` 命令进行全文搜索, 如果要查询中文内容的话, 需要在添加数据时设置中文编码, 并且在查询时也要设置中文编码, 指定“`language "chinese"`”。

参考 & 鸣谢

官网地址:

<http://redisearch.io>

项目地址：

<https://github.com/RediSearch/RediSearch>

[上一页](#)

[下一页](#)