

Redis与网站架构

@joyqi
SegmentFault.com

什么是Redis?

什么是Redis?

- REmote DIctionary Server 缩写
- 一个基于内存的网络存储系统
- 丰富的数据结构(sets, sorted sets, hashes, list ...)
- 本质是key-value，但是与memcached不同的是，value的类型得到了扩展

有何不同？
看看具体的网站

一个普通的问题列表需求

- 问题本身的数据(标题, 投票等等)
- 问题的作者数据(另一张单独的一张数据表, 通过某个键值关联)
- 问题的标签(本身单独一张数据表, 通过一个中间关系表与问题产生一对多的关系)



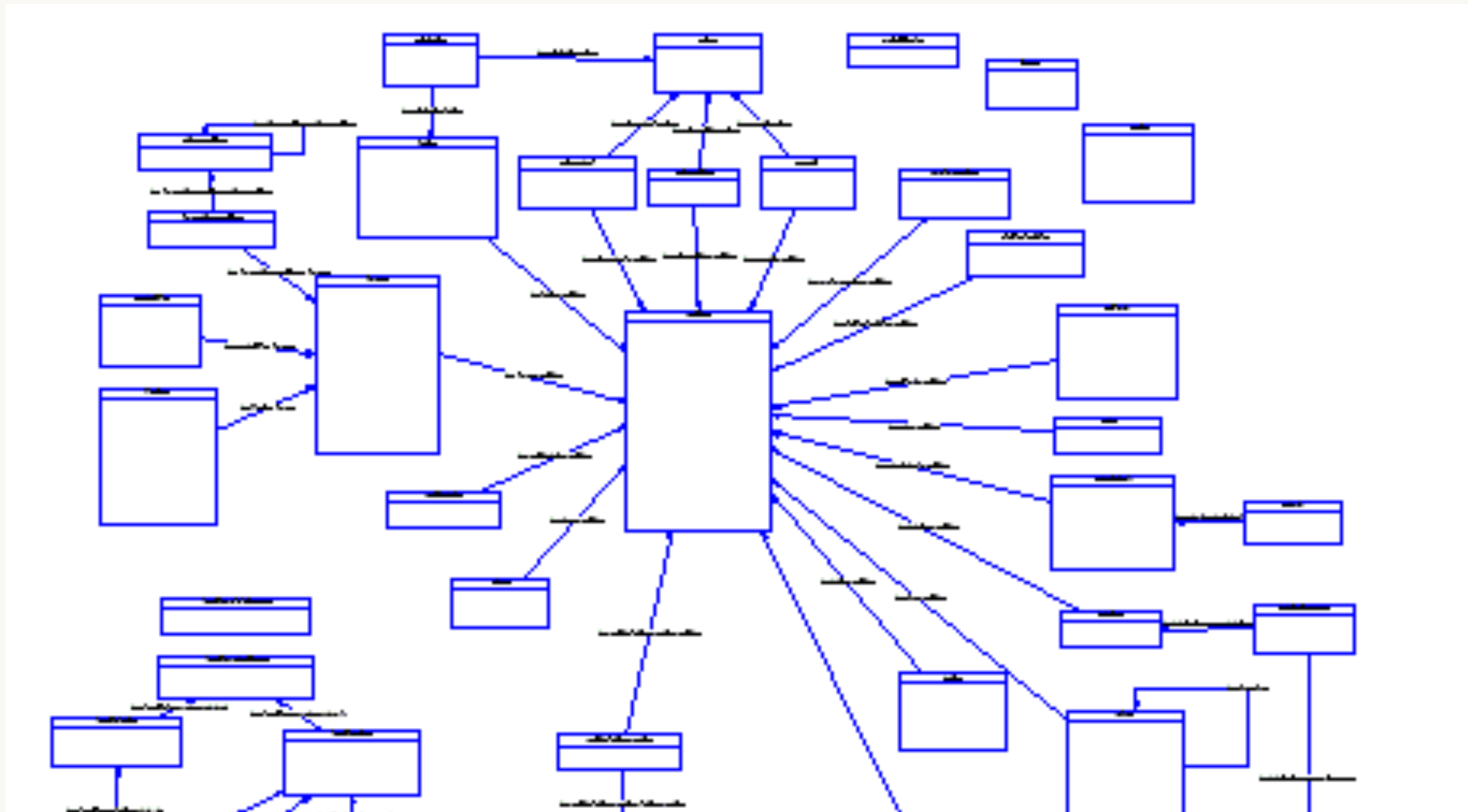
如何取出数据？

SELECT ...

SQL无法回避的问题


```
select www_name,user_pw,c_name,join_time,user_ad,end_time,user_id,user_type,user_tel,user_email_one,birthday from
user where www_name="111111" and c_name="111111" and birthday="1980-07-01" and user_ad="1111111" and
user_tel="11111111" and user_email_one="111@111.111" and login_time="11" and join_time>="2007-07-29" and
join_time<="" and user_living=2 and user_living=1 and user_id in(select user_id from (select
user_id,login_num/(period_diff(extract(year_month from now()),extract(year_month from join_time))+1) as piju from user
group by user_id) user where piju>=1 and piju<=20) and user_id in(select user_id from (select user_id,year(now())-
year(birthday) as nl from user group by user_id) user where nl>=20 and nl<=40) and user_ad like "%111111%" and
user_id in(select user_id from (select user_id,count(goods_id) as mj from goods where end_time>="2007-05-29" and
end_time<="2008-01" group by user_id) goods where mj>=10 and mj<=20) and user_id in(select user_id from (select
user_id,count(goods_id) as cgmj from goods where end_time>="2007-05-29" and end_time<="2008-01" and g_ok=1
group by user_id) goods where cgmj>=1 and cgmj<=20) and user_id in(select user_id from (select
user_id,count(goods_id) as sbmj from goods where end_time>="2007-05-29" and end_time<="2008-01" and g_ok=2
group by user_id) goods where sbmj>=1 and sbmj<=20) and user_id in(select user_id from (select distinct user_id,(select
sum(goods_total) as zs from goods where user_id=g.user_id and end_time>="2007-05-29" and end_time<="2008-01"
group by user_id) as zs,(select sum(goods_total) as cg from goods where user_id=g.user_id and end_time>="2007-05-
29" and end_time<="2008-01" and g_ok=1 group by user_id)as cg from goods as g) as g where cg/zs>=0.001 and
cg/zs<=0.9) and user_id in(select user_id from (select distinct user_id,(select sum(goods_total) as zs from goods where
user_id=g.user_id and end_time>="2007-05-29" and end_time<="2008-01" group by user_id) as zs,(select
sum(goods_total) as sb from goods where user_id=g.user_id and end_time>="2007-05-29" and end_time<="2008-01"
and g_ok=2 group by user_id)as sb from goods as g) as g where sb/zs>=0.01 and sb/zs<=0.9) and user_id in(select
```

用一条SQL解决问题是不可能的



多次查询会让大家崩溃



冗余字段会让你看起来很傻

我看还是用Memcached好了
等等,为啥不试试Redis

用Hashes保存字段

```
$user = array(  
    'id'      => 123,  
    'name'    => 'joyqi',  
    'mail'    => 'magike.net@gmail.com',  
    'created' => 1212312312  
);
```

```
$redis->hMSet('user:123', $user);  
print_r($redis->hGetAll('user:123'));
```

用Sets保存关系

```
$questionId = 123;  
$tagIds = array(111, 222, 333);  
foreach ($tagIds as $sort => $tagId) {  
    $redis->zAdd('question_tag:' . $questionId,  
$sort, $tagId);  
}  
  
print_r($redis->zRange('question_tag:123', 0,  
-1));
```

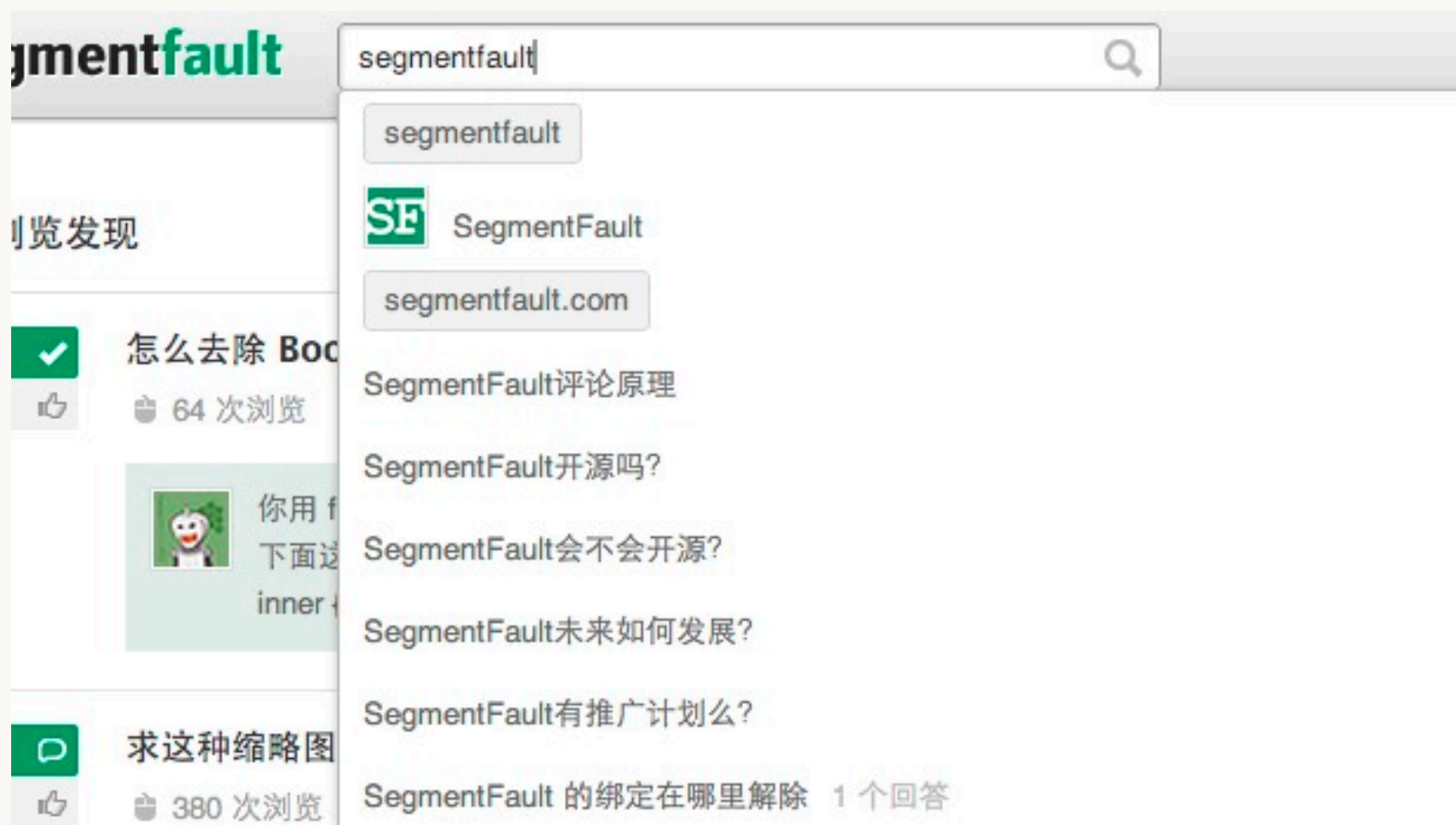
与SQL相比

- 大大减少了查询数量，提高了效率
- redis的API更加人性化，再也不需要构建SQL语句，节省了SQL的解析时间

与Memcached相比

- 支持哈希存储，存储多个字段时不需要自己再次用类似json的格式编码和解码，更新也更加高效
- 支持列表和集合等多种数据集，可以方便快捷地处理有序数据

Redis 用于搜索?



有了Redis，支持实时搜索的门槛被大大降低

分词加Redis集合实现索引

```
$questionTitle = '搜索技术';  
$questionId = 123;  
$words = fenci($questionTitle); // array('搜索', '技术');  
foreach ($words as $word) {  
    $redis->zAdd('w:' . md5($word), 1, $questionId);  
}  
  
// 索引完成
```

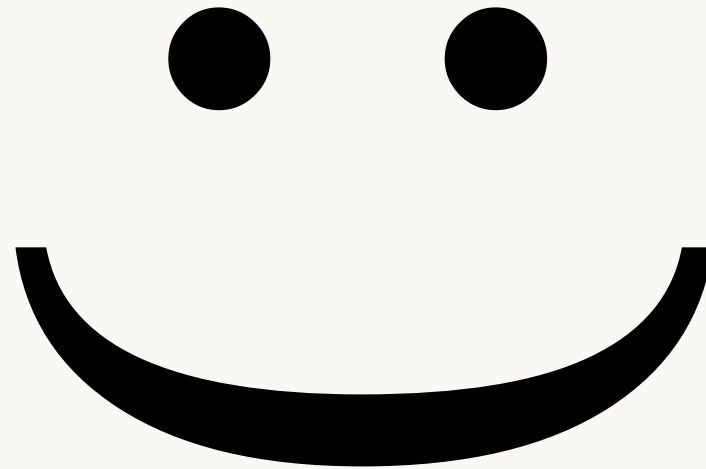
Redis集合Union操作实现查询

```
$keywords = '怎样实现搜索技术';  
$words = fenci($keywords); // array('怎样', '实现', '搜索', '技术');  
  
$indexes = array_map(function ($word) {  
    return 'w:' . md5($word);  
}, $words);  
  
$redis->zUnion('result', $indexes, array_fill(0, count($indexes), 1),  
    'sum');  
print_r($redis->zRevRange('result', 0, -1));
```


深远影响

Redis的特性是如此丰富而又恰到好处，他会给网站架构带来深远影响，一些以前无法实现的方法将涌现出来。

<http://redis.io>



Thanks, Q&A

@joyqi | joyqi.com