主讲人：付磊

主页：http://carlosfu.iteye.com/

4-3.哈希

哈希

特点

“重要”API和实战

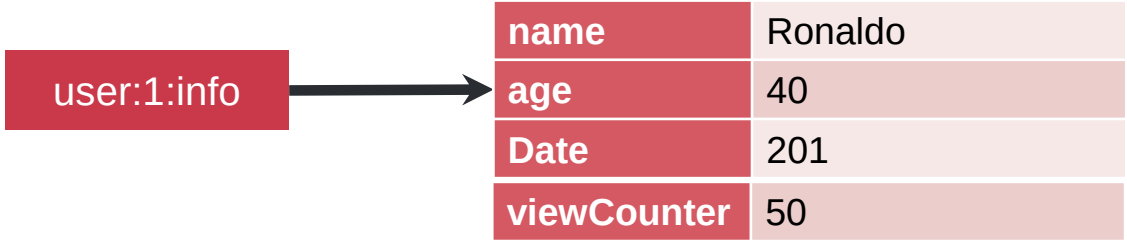
hash vs string

查缺补漏

1.哈希

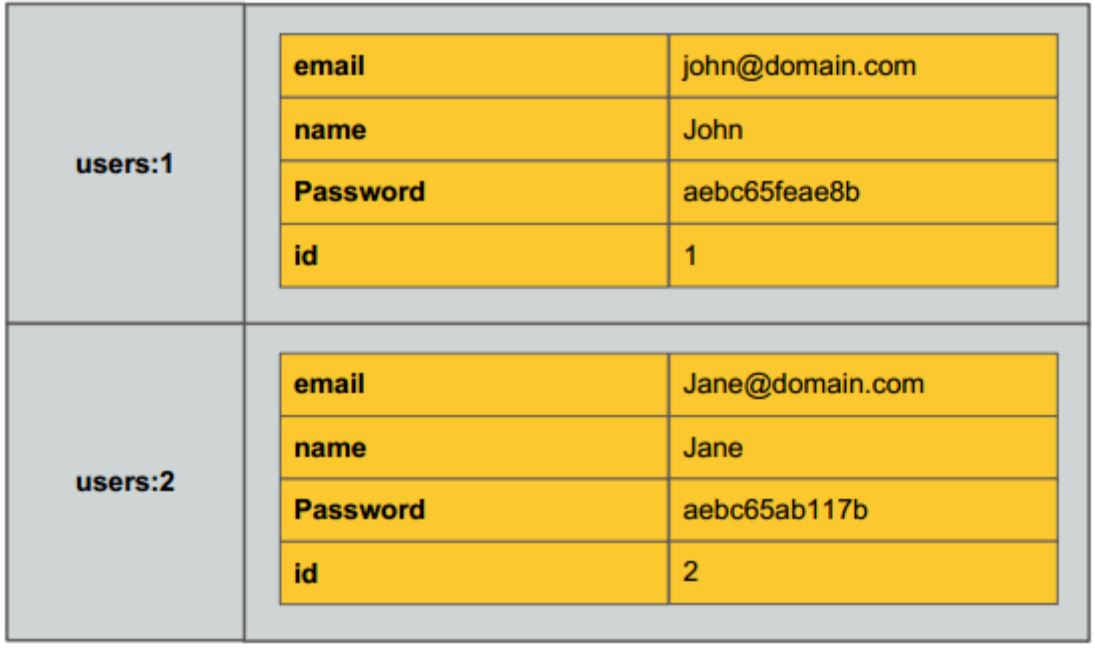
哈希键值结构

key　　field　　value



add a new value

viewCounter	50



特点

Mapmap?

Small redis

field不能相同,value可以相同

2. “重要”API

H

hget、hset、hdel

API

hget key field
#获取hash key对应的field的value

o(1)

API

hset key field value
#设置hash key对应field的value

o(1)

API

hdel key field
#删除hash key对应field的value

o(1)

hget、hset、hdel

演示

```
127.0.0.1:6379> hset user:1:info age 23
(integer) 1
127.0.0.1:6379> hget user:1:info age
"23"
127.0.0.1:6379> hset user:1:info name ronaldo
(integer) 1
127.0.0.1:6379> hgetall user:1:info
1) "age"
2) "23"
3) "name"
4) "ronaldo"
127.0.0.1:6379> hdel user:1:info age
(integer) 1
127.0.0.1:6379> hgetall user:1:info
1) "name"
2) "ronaldo"
```

jedis

```
Jedis jedis = new Jedis(“127.0.0.1", 6379);
String key = "user:1:info";
// 设置field和value
jedis.hset(key, "age", "23");
jedis.hset(key, "name", "ronaldo");
Map<String, String> userInfoMap = jedis.hgetAll(key);
System.out.println("key: " + key + ", value is " + userInfoMap);
```

hexists、hlen

API

hexists key field
#判断hash key是否有field

o(1)

API

hlen key
#获取hash key field的数量

o(1)

hexists、hlen

演示

```
127.0.0.1:6379> hgetall user:1:info
1) "name"
2) "ronaldo"
3) "age"
4) "23"
127.0.0.1:6379> hexists user:1:info name
(integer) 1
127.0.0.1:6379> hlen user:1:info
(integer) 2
```

hmget、hmset

API

hmget key field1 field2…. fieldN
#批量获取hash key的一批field对应的值

o(n)

API

hmset key field1 value1 field2 value2…fieldN valueN
#批量设置hash key的一批field value

o(n)

hmget、hmset

演示

```
127.0.0.1:6379> hmset user:2:info age 30 name kaka page 50
OK
127.0.0.1:6379> hlen user:2:info
(integer) 3
127.0.0.1:6379> hmget user:2:info age name
1) "30"
2) "kaka"
```

实战！

实现如下功能：

记录网站每个用户个人主页的访问量？

hincrby user:1:info pageview count

实战！

实现如下功能：

缓存视频的基本信息（数据源在mysql中）伪代码

```
public VideoInfo get(long id) {
        String redisKey = redisPrefix + id;
        Map<String,String> hashMap = redis.hgetAll(redisKey);
        VideoInfo videoInfo = transferMapToVideo(hashMap);
        if (videoInfo == null) {
            videoInfo = mysql.get(id);
            if (videoInfo != null) {
                redis.hmset(redisKey, transferVideoToMap(videoInfo));
            }
        }
        return videoInfo;
    }
```

hgetall、hvals、hkeys

API

hgetall key
#返回hash key对应所有的field和value

o(n)

API

hvals key
#返回hash key对应所有field的value

o(n)

API

hkeys key
#返回hash key对应所有field

o(n)

hgetall、hvals、hkeys

演示

```
127.0.0.1:6379> hgetall user:2:info
1) "age"
2) "30"
3) "name"
4) "kaka"
5) "page"
6) "50"
127.0.0.1:6379> hvals user:2:info
1) "30"
2) "kaka"
3) "50"
127.0.0.1:6379> hkeys user:2:info
1) "age"
2) "name"
3) "page"
```

小心使用hgetall

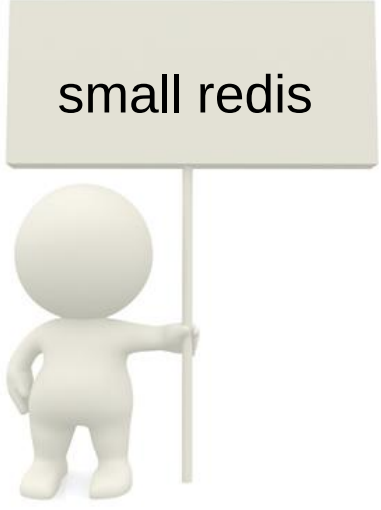
小心使用
牢记单线程

3. string vs hash

相似的API

get set setnx del incr incrby decr decrby mset mget	hget hset hsetnx hdel hincrby hmset hmget

哈希



用户信息(string实现)

key

value(serializable:json,xml,protobuf)

user:1


```
{
  “id”:1
  “name": “ronaldo",
  “age": 40,
  “pageView": 500000
}
```

set user:1 serialize(userinfo)

```
{
  “id”:1
  “name": “carlos",
  “age": 42,
  “pageView": 800000
}
```

用户信息(string实现-v2)

41

set user:1:age 41

set user:1:link tv.sohu.com

key	value
user:1:name	world
user:1:age	40
user:1:pageView	500000
user:1:link	tv.sohu.com

用户信息(hash)

key　　field　　value

user:1:info

name	ronaldo
age	40
pageView	500000
link	tv.sohu.com

41

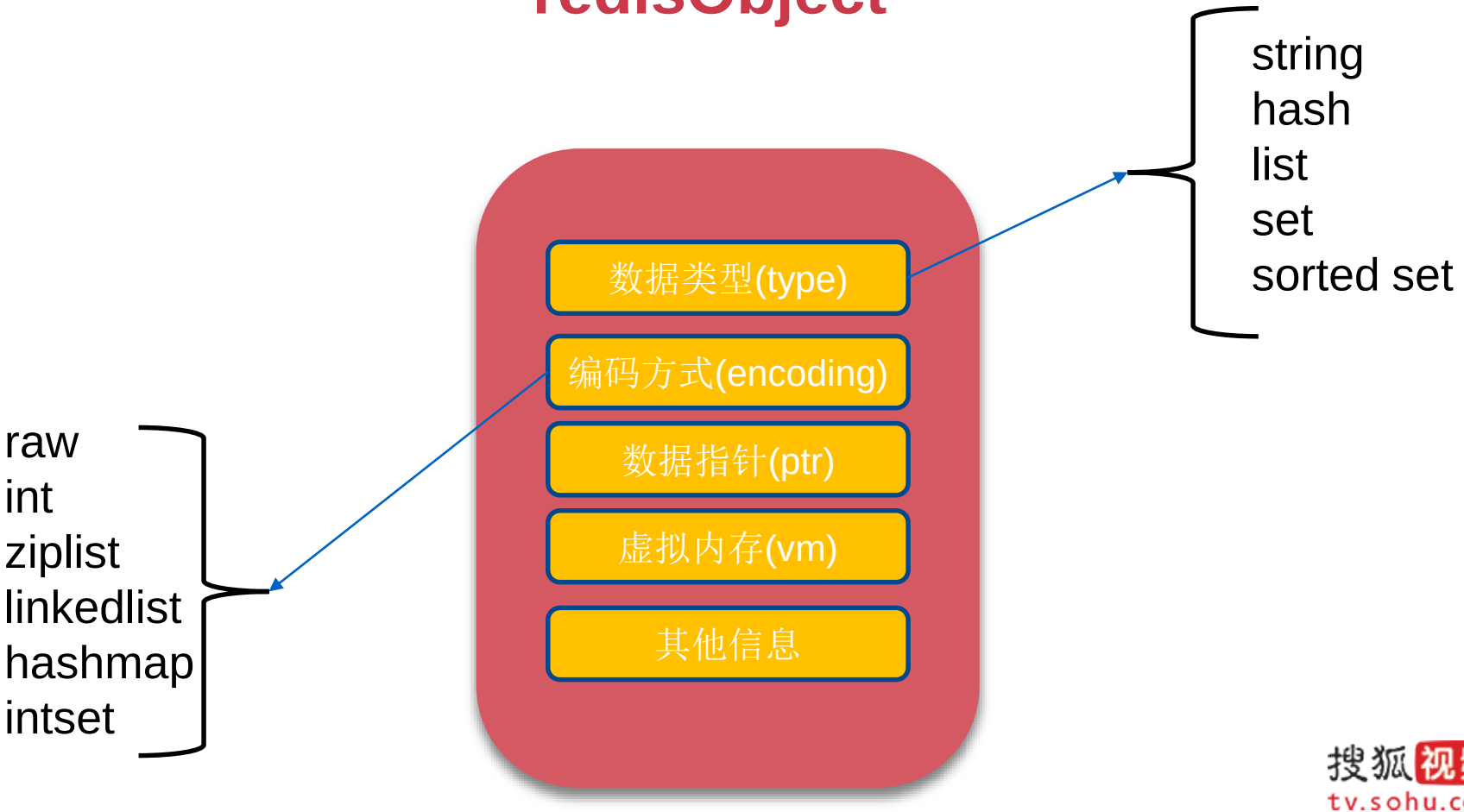
hset user:1:info age 41

hset user:1:info link tv.sohu.com

3种方案比较

命令	优点	缺点
string v1	编程简单 可能节约内存	1. 序列化开销 2. 设置属性要操作整个数据。
string v2	直观 可以部分更新	1. 内存占用较大 2. key较为分散
hash	直观 节省空间 可以部分更新	1. 编程稍微复杂 2. ttl不好控制

redisObject



节省内存

1. Instagram:21G->5G
2. 美团内存优化

4. 查缺补漏

hsetnx、hincrby、hincrbyfloat

api

hsetnx key field value

#设置hash key对应field的value(如field已经存在，则失败)

o(1)

api

hincrby key field intCounter

#hash key对应的field的value自增intCounter

o(1)

api

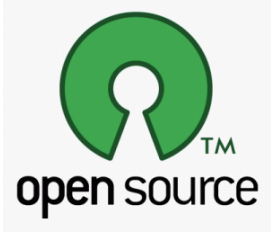
hincrbyfloat key field floatCounter

#hincrby浮点数版

o(1)

哈希总结

命令	复杂度
hget hset hdel	o(1)
hexists	o(1)
hincrby	o(1)
hgetall hvals hkeys	o(n)
hmget hmset	o(n)



搜狐视频Redis私有云平台开源了！！

Github主页：https://github.com/sohutv/cachecloud

QQ群：534429768