



主讲人：付磊

主页：<http://carlosfu.iteye.com/>

## 4-2.字符串



# 字符串

字符串键值

“重要”API和实战

查缺补漏

# 1. 字符串键值

# 字符串键值结构

key

value

hello

world

counter

1

bits

1

0

1

1

1

1

0

1

```
{  
  "product": {  
    "id": "2951",  
    "name": "testing 01",  
    "quantity": 4  
  }  
}
```

※ Up to 512MB

# 场景

缓存

计数器

分布式锁

.....

## 2. “重要” API

# get、set、del

API

get key  
#获取key对应的value

$O(1)$

API

set key value  
#设置key-value

$O(1)$

API

del key  
#删除key-value

$O(1)$



# get、set、del

演示

```
redis> set hello "world"
```

```
OK
```

```
redis> get hello
```

```
"world"
```

```
redis> del hello
```

```
1
```

```
redis> get hello
```

```
nil
```

# jedis

```
Jedis jedis = new Jedis("127.0.0.1", 6379);  
String key = "hello";  
// 设置  
String setResult = jedis.set(key, "world");  
System.out.println("jedis setResult is " + setResult);  
// 获取  
String getResult = jedis.get(key);  
System.out.println("jedis getResult is " + getResult);  
// 删除  
Long delResult = jedis.del(key);  
System.out.println("jedis delResult is " + delResult);  
// 重新获取  
String getNewResult = jedis.get("hello");  
System.out.println("jedis getNewResult is " + getNewResult);
```

# Incr、decr、Incrby、decrby

API

incr key

#key自增1,如果key不存在,自增后get(key)=1

$O(1)$

API

decr key

#key自减1,如果key不存在,自减后get(key)=-1

$O(1)$

API

incrby key k

#key自增k,如果key不存在,自增后get(key)=k

$O(1)$

API

decr key k

#key自减k,如果key不存在,自减后get(key)=-k

$O(1)$

# Incr、decr、Incrby、decrby

演示

```
127.0.0.1:6379> get counter
```

```
(nil)
```

```
127.0.0.1:6379> incr counter
```

```
(integer) 1
```

```
127.0.0.1:6379> get counter
```

```
"1"
```

```
127.0.0.1:6379> incrby counter 99
```

```
(integer) 100
```

```
127.0.0.1:6379> decr counter
```

```
(integer) 99
```

```
127.0.0.1:6379> get counter
```

```
"99"
```

# 实战！

实现如下功能：

记录网站每个用户个人主页的访问量？



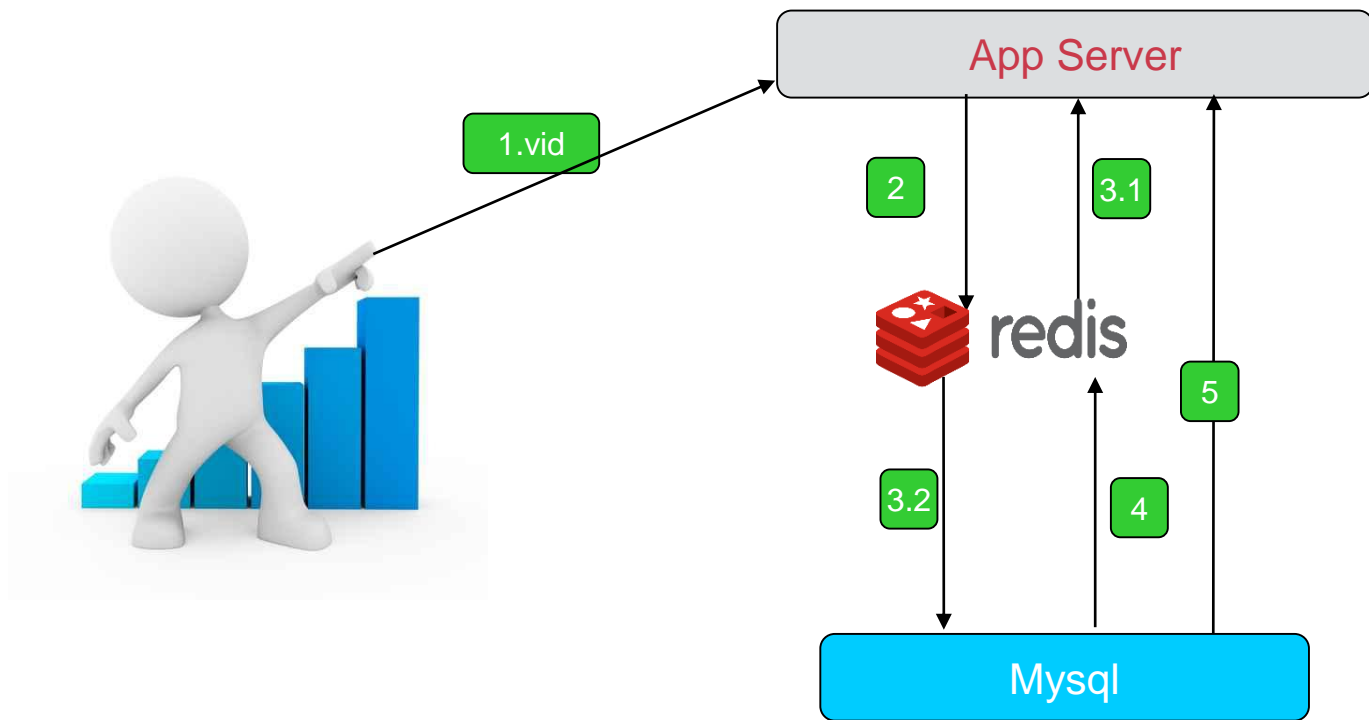
incr userid:pageview

# 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







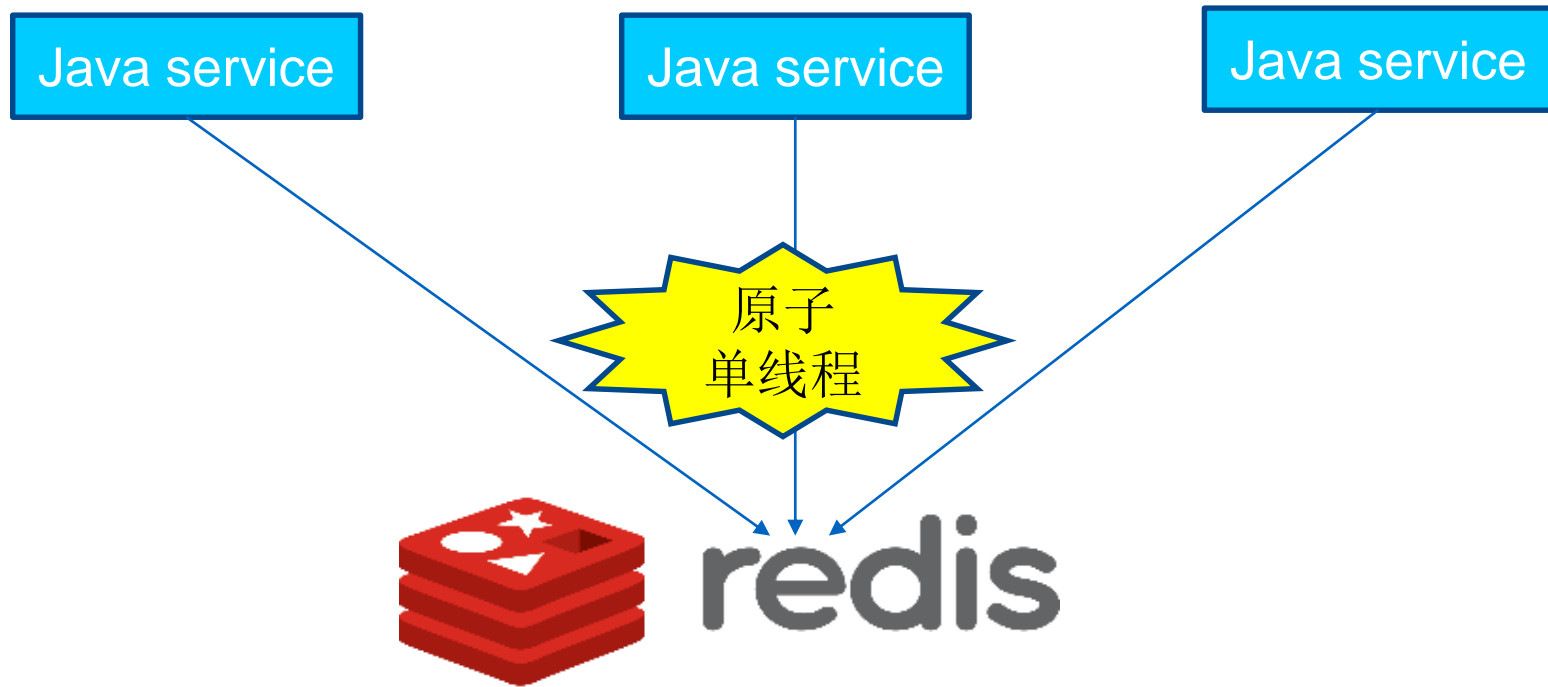
```
public VideoInfo get(long id) {  
    String redisKey = redisPrefix + id;  
    VideoInfo videoInfo = redis.get(redisKey);  
    if (videoInfo == null) {  
        videoInfo = mysql.get(id);  
        if (videoInfo != null) {  
            //序列化  
            redis.set(redisKey, serialize(videoInfo));  
        }  
    }  
    return videoInfo;  
}
```

# 实战！

实现如下功能：

分布式id生成器





incr id (原子操作)

# set setnx setxx

API

set key value  
#不管key是否存在，都设置

$O(1)$

API

setnx key value  
#key不存在，才设置

$O(1)$

API

set key value xx  
#key存在，才设置

$O(1)$

# set setnx set xx

## 演示

```
127.0.0.1:6379> exists php
(integer) 0
127.0.0.1:6379> set php good
OK
127.0.0.1:6379> setnx php bad
(integer) 0
127.0.0.1:6379> set php best xx
OK
127.0.0.1:6379> get php
"best"
127.0.0.1:6379> exists java
(integer) 0
127.0.0.1:6379> setnx java best
(integer) 1
127.0.0.1:6379> set java easy xx
OK
127.0.0.1:6379> get java
"easy"
127.0.0.1:6379> exists lua
(integer) 0
127.0.0.1:6379> set lua hehe xx
(nil)
```

# mget mset

API

mget key1 key2 key3...  
#批量获取key,原子操作

$O(n)$

API

mset key1 value1 key2 value2 key3 value3  
#批量设置key-value

$O(n)$

## mget mset

```
127.0.0.1:6379> mset hello world java best php good
```

```
OK
```

```
127.0.0.1:6379> mget hello java php
```

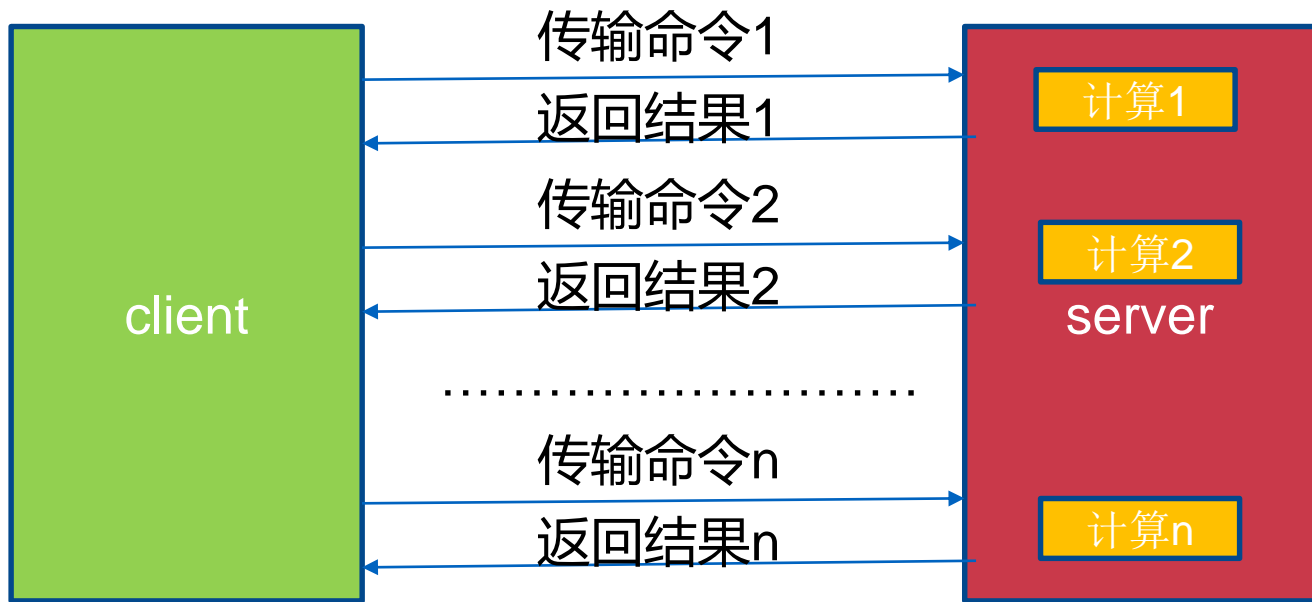
```
1) "world"
```

```
2) "best"
```

```
3) "good"
```



## n次get



$$n\text{次get} = n\text{次网络时间} + n\text{次命令时间}$$

# 1次mget



$$1\text{次mget} = 1\text{次网络时间} + n\text{次命令时间}$$

### 3. 查缺补漏

# getset、append、strlen

api

getset key newvalue  
#set key newvalue并返回旧的value

$O(1)$

api

append key value  
#将value追加到旧的value

$O(1)$

api

strlen key  
#返回字符串的长度(注意中文)

$O(1)$

# getset、append、strlen

演示

```
127.0.0.1:6379> set hello world
```

```
OK
```

```
127.0.0.1:6379> getset hello php
```

```
"world"
```

```
127.0.0.1:6379> append hello ",java"
```

```
(integer) 8
```

```
127.0.0.1:6379> get hello
```

```
"php,java"
```

```
127.0.0.1:6379> strlen hello
```

```
(integer) 8
```

```
127.0.0.1:6379> set hello "足球"
```

```
OK
```

```
127.0.0.1:6379> strlen hello
```

```
(integer) 4
```

# Incrbyfloat getrange setrange

api

Incrbyfloat key 3.5  
#增加key对应的值3.5

$O(1)$

api

getrange key start end  
#获取字符串指定下标所有的值

$O(1)$

api

setrange key index value  
#设置指定下标所有对应的值

$O(1)$

# Incrbyfloat getrange setrange

演示

```
127.0.0.1:6379> incr counter
(integer) 1
127.0.0.1:6379> incrbyfloat counter 1.1
"2.1"
127.0.0.1:6379> get counter
"2.1"
127.0.0.1:6379> set hello javabest
OK
127.0.0.1:6379> getrange hello 0 2
"jav"
127.0.0.1:6379> setrange hello 4 p
(integer) 8
127.0.0.1:6379> get hello
"javapest"
```

# 字符串总结

命令	含义	复杂度
set key value	设置key-value	$O(1)$
get key	获取key-value	$O(1)$
del key	删除key-value	$O(1)$
setnx setxx	根据key是否存在设置key-value	$O(1)$
Incr decr	计数	$O(1)$
mget mset	批量操作key-value	$O(n)$



# 字符串总结

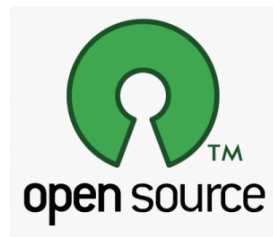
缓存

计数器

分布式锁

.....





搜狐视频Redis私有云平台开源了！！

Github主页: <https://github.com/sohutv/cacheccloud>

QQ群: 534429768