



链滴

redis 命令 exists vs get

作者: [zeekling](#)

原文链接: <https://ld246.com/article/1668097328943>

来源网站: 链滴

许可协议: [署名-相同方式共享 4.0 国际 \(CC BY-SA 4.0\)](#)

功能对比

exists

参数格式:

EXISTS key [key ...]

用于判断某个键是否存在

get

参数格式:

GET key

用于获取键对应的值。

由上可知，相同的场景只有判断键是否存在。

性能对比

使用下面命令向Redis里面注入大量数据:

```
public class FillData {  
  
    private static final Jedis jedis = new Jedis("127.0.0.1", 6379);  
  
    private static void fillData(long min, long max, SetParams params) {  
        for (long i = min; i < max; i++) {  
            jedis.set("key_0000000" + i, "value_0000000" + i, params);  
            if (i % 1000 == 0) {  
                System.out.println("count=" + i);  
            }  
        }  
    }  
  
    public static void main(String[] args) {  
        SetParams params = new SetParams();  
        FillData.fillData(0, 5000000, params);  
        params.ex(50000L);  
        FillData.fillData(50000000, 55000000, params);  
    }  
}
```

使用下面代码对比相应时间:

```
public class ExistsVSGet {  
  
    private static final Jedis jedis = new Jedis("127.0.0.1", 6379);
```

```

private static void exists() {
    long begin = System.nanoTime();
    jedis.exists("key_aaaaaaa");
    long end = System.nanoTime();
    System.out.println("exists cost=" + (end - begin)/1000);
}
private static void get() {
    long begin = System.nanoTime();
    jedis.exists("key_aaaaaaa");
    long end = System.nanoTime();
    System.out.println("get cost=" + (end - begin)/1000);
}

private static void test() {
    for (int i=1; i< 10; i++) {
        jedis.get("0000000" + i);
    }
}

public static void main(String[] args) {
    ExistsVSGet.test();
    ExistsVSGet.get();
    ExistsVSGet.exists();
}
}

```

执行结果如下：

```

get cost=476
exists cost=279

```

由此可见，get性能要比exists差。

源码分析

get命令核心查找key代码

```

robj *lookupKeyReadOrReply(client *c, robj *key, robj *reply) {
    robj *o = lookupKeyRead(c->db, key);
    if (!o) addReply(c,reply);
    return o;
}

```

其中lookupKeyRead的实现：

```

robj *lookupKeyRead(redisDb *db, robj *key) {
    return lookupKeyReadWithFlags(db,key,LOOKUP_NONE);
}

```

exists命令核心实现：

```

void existsCommand(client *c) {
    long long count = 0;

```

```

int j;

for (j = 1; j < c->argc; j++) {
    if (lookupKeyReadWithFlags(c->db,c->argv[j],LOOKUP_NOTOUCH)) count++;
}
addReplyLongLong(c,count);
}

```

exists和get命令都调用了lookupKeyReadWithFlags，我们看下这个函数的实现：

```

robj *lookupKeyReadWithFlags(redisDb *db, robj *key, int flags) {
    robj *val;

    if (expireIfNeeded(db,key) == 1) {
        if (server.masterhost == NULL)
            goto keymiss;
        if (server.current_client &&
            server.current_client != server.master &&
            server.current_client->cmd &&
            server.current_client->cmd->flags & CMD_READONLY)
        {
            goto keymiss;
        }
    }
    val = lookupKey(db,key,flags);
    if (val == NULL)
        goto keymiss;
    server.stat_keyspace_hits++;
    return val;

keymiss:
    if (!(flags & LOOKUP_NONOTIFY)) {
        server.stat_keyspace_misses++;
        notifyKeyspaceEvent(NOTIFY_KEY_MISS, "keymiss", key, db->id);
    }
    return NULL;
}

```

可见都是先判断了确认了是否过期(过期key单独保存在另一个dict里面)，再没有过期的情况下查找了d库，总体查找时间是一致的。

但是，我们可以发现，get命令返回了键对应的值，exists返回了个数，一般键对应的值较大，传输时较长。所以相对较慢。

结论

在判断key是否存在的场景下：

- exists速度更快，可以忽略类型。
- get只适用于string类型，当值越大时，速度越慢。