Please visit website: http://cxyroad.com

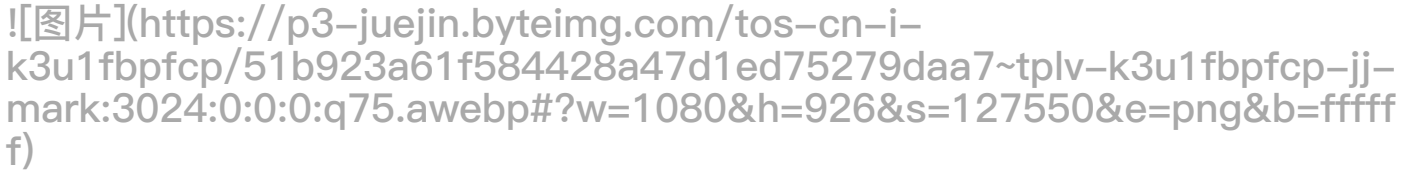
工作中用Redis最多的10种场景

前言

> Java突击队：[www.susan.net.cn](http://cxyroad.com/”https://link.zhihu.com/?target=http%3A//www.susan.net.cn”)，一个能够帮你快速提升java技术的神奇网站。

Redis作为一种优秀的基于key/value的缓存，有非常不错的性能和稳定性，无论是在工作中，还是面试中，都经常会出现。

今天这篇文章就跟大家一起聊聊，我在实际工作中使用Redis的10种场景，希望对你会有所帮助。



[8000页BAT大佬写的刷题笔记，让我offer拿到手软](http://cxyroad.com/”http://www.susan.net.cn/zt/8000.html”)

1. 统计访问次数

对于很多官方网站的首页，经常会有一些统计首页访问次数的需求。

访问次数只有一个字段，如果保存到数据库中，再最后做汇总显然有些麻烦。

该业务场景可以使用Redis，定义一个key，比如：OFFICIAL_INDEX_VISIT_COUNT。

在Redis中有incr命令，可以实现给value值加1操作：

```
incr OFFICIAL_INDEX_VISIT_COUNT
```

当然如果你想一次加的值大于1，可以用incrby命令，例如：

```
incrby OFFICIAL_INDEX_VISIT_COUNT 5
```

这样可以一次性加5。

2. 获取分类树

在很多网站都有分类树的功能，如果没有生成静态的html页面，想通过调用接口的方式获取分类树的数据。

我们一般为了性能考虑，会将分类树的json数据缓存到Redis当中，为了后面在网站当中能够快速获取数据。

不然在接口中需要使用递归查询数据库，然后拼接成分类树的数据结构。

这个过程非常麻烦，而且需要多次查询数据库，性能很差。

因此，可以考虑用一个定时任务，异步将分类树的数据，直接缓存到Redis当中，定义一个key，比如：MALL_CATEGORY_TREE。

然后接口中直接使用MALL_CATEGORY_TREE这个key从缓存中获取数据即可。

可以直接用key/value字符串保存数据。

不过需要注意的是，如果分类树的数据非常多可能会出现大key的问题，优化方案可以参考我的另外一篇文章《[分类树，我从2s优化到0.1s](http://cxyroad.com/”https://mp.weixin.qq.com/s?__biz=MzkwNjMwMTgzMQ==&mid=2247504097&idx=1&sn=1cac86e9b9152a93e4bfca56c3b9bf18&chksm=c0e80809f79f811f703408480a868967a5be3c085ac6fef5b6de7fc5e235efa3965d343965c3&token=795379312&lang=zh_CN&scene=21#wechat_redirect”)》。

3. 做分布式锁

分布式锁可能是使用Redis最常见的场景之一，相对于其他的分布式锁，比如：数据库分布式锁或者Zookeeper分布式锁，基于Redis的分布式锁，有更好的性能，被广泛使用于实际工作中。

我们使用下面这段代码可以加锁：

```
try{
  String result = jedis.set(lockKey, requestId, ”NX”, ”PX”, expireTime);
  if (”OK”.equals(result)) {
      return true;
  }
  return false;
} finally {
    unlock(lockKey);
}
```

但上面这段代码在有些场景下，会有一些问题，释放锁可能会释放了别人的锁。

说实话Redis分布式锁虽说很常用，但坑也挺多的，如果用不好的话，很容易踩坑。

如果大家对Redis分布式锁的一些坑比较感兴趣，可以看看我的另一篇文章《[聊聊redis分布式锁的8大坑](http://cxyroad.com/

”https://mp.weixin.qq.com/s?__biz=MzkwNjMwMTgzMQ==&mid=2247490430&idx=1&sn=a1f42f9a981a8f161941a6472f317b10&chksm=c0ebc396f79c4a801a330917ca700e7d7a6af3a3c2c5a4e11a05770da925de8aa9ed3c277737&token=795379312&lang=zh_CN&scene=21#wechat_redirect”)》，文章中有非常详细的介绍。

4. 做排行榜

很多网站有排行榜的功能，比如：商城中有商品销量的排行榜，游戏网站有玩家获得积分的排行榜。

通常情况下，我们可以使用`Sorted Set`保存排行榜的数据。

使用`ZADD`可以添加排行榜的数据，使用`ZRANGE`可以获取排行榜的数据。

例如：

```
ZADD rank:score 100 ”周星驰”
ZADD rank:score 90 ”周杰伦”
ZADD rank:score 80 ”周润发”
ZRANGE rank:score 0 -1 WITHSCORES
```

返回数据：

```
1) ”周星驰”
2) ”100”
3) ”周杰伦”
4) ”90”
5) ”周润发”
6) ”80”
```

5. 记录用户登录状态

通常下，用户登录成功之后，用户登录之后的状态信息，会保存到Redis中。

这样后面该用户访问其他接口的时候，会直接从Redis中查询用户登录状态，如果可以查到数据，说明用户已登录，则允许做后续的操作。

如果从Redis中没有查到用户登录状态，说明该用户没有登录，或者登录状态失效了，则直接跳转到用户登录页面。

使用Redis保存用户登录状态，有个好处是它可以设置一个过期时间，比如：该时间可以设置成30分钟。

```
jedis.set(userId, userInfo, 1800);
```

在Redis内部有专门的job，会将过期的数据删除，也有获取数据时实时删除的逻辑。

6. 限流

使用Redis还有一个非常常用的的业务场景是`做限流`。

当然还有其他的限流方式，比如：使用nginx，但使用Redis控制可以更精细。

比如：限制同一个ip，1分钟之内只能访问10次接口，10分钟之内只能访问50次接口，1天之内只能访问100次接口。

如果超过次数，则接口直接返回：请求太频繁了，请稍后重试。

跟上面保存用户登录状态类似，需要在Redis中保存用户的请求记录。

比如：key是用户ip，value是访问的次数从1开始，后面每访问一次则加1。

如果value超过一定的次数，则直接拦截这种异常的ip。

当然也需要设置一个过期时间，异常ip如果超过这个过期时间，比如：1天，则恢复正常了，该ip可以再发起请求了。

或者限制同一个用户id。

7. 位统计

比如现在有个需求：有个网站需要统计一周内连续登陆的用户，以及一个月内登陆过的用户。

这个需求使用传统的数据库，实现起来比较麻烦，但使用Redis的`bitmap`让我们可以实时的进行类似的统计。

bitmap 是二进制的byte数组，也可以简单理解成是一个普通字符串。它将二进制数据存储在byte数组中以达到存储数据的目的。

保存数据命令使用setbit，语法：

```
setbit key offset value
```

具体示例：

```
setbit user:view:2024-01-17 123456 1
```

往bitmap数组中设置了用户id=123456的登录状态为1，标记2024-01-17已

登录。

然后通过命令getbit获取数据，语法：

```
getbit key offset
```

具体示例：

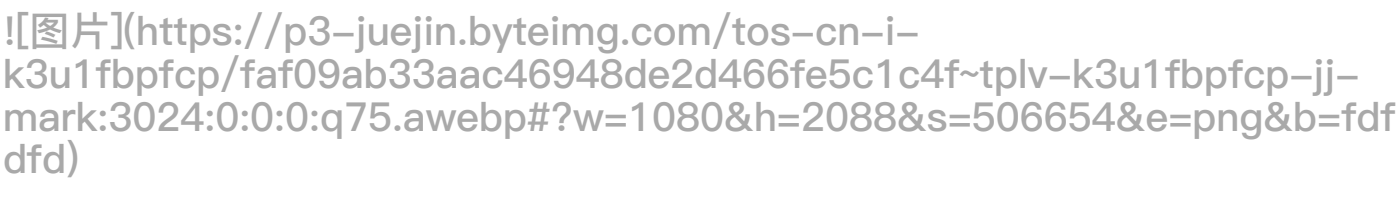
```
getbit user:view:2024-01-17 123456
```

如果获取的值是1，说明这一天登录了。

如果我们想统计一周内连续登录的用户，只需要遍历用户id，根据日期中数组中去查询状态即可。

最近就业形式比较困难，为了感谢各位小伙伴对苏三一直以来的支持，我特地创建了一些工作内推群， 看看能不能帮助到大家。

你可以在群里发布招聘信息，也可以内推工作，也可以在群里投递简历找工作，也可以在群里交流面试或者工作的话题。



进群方式

添加，苏三的**私人**：su_san_java，备注：**内推+所在城市**，即可加入。

8. 缓存加速

我们在工作中使用Redis作为缓存加速，这种用法也是非常常见的。

如果查询订单数据，先从Redis缓存中查询，如果缓存中存在，则直接将数据返回给用户。

如果缓存中不存在，则再从数据库中查询数据，如果数据存在，则将数据保存到缓存中，然后再返回给用户。

如果缓存和数据库都不存在，则直接给用户返回数据不存在。

流程图如下：![图片](https://p3-juejin.byteimg.com/tos-cn-i-k3u1fbpfcp/234b86c5c6284ac3bb7a04a0c9e6468d~tplv-k3u1fbpfcp-jj-mark:3024:0:0:0:q75.awebp#?w=980&h=776&s=128019&e=png&b=fffefe)但使用缓存加速的业务场景，需要注意一下，可能会出现：缓存击穿、穿透和雪崩等问题，感兴趣的小伙伴，可以看看我的另一篇文章《[烂大街的缓存穿透、缓存击穿和缓存雪崩，你真的懂了？](http://cxyroad.com/”https://mp.weixin.qq.com/s?__biz=MzkwNjMwMTgzMQ==&mid=2247491225&idx=1&sn=bfb14f28911efaa6e3a615870fff9a5c&chksm=c0ebc671f79c4f6718a63bbec91d79a4b05e1dd2a7c00ed9bf6a9f582abd4c6e5c870148ff89&token=795379312&lang=zh_CN&scene=21#wechat_redirect”)》，里面有非常详细的介绍。

9. 做消息队列

我们说起队列经常想到是：kafka、rabbitMQ、RocketMQ等这些分布式消息队列。

其实Redis也有消息队列的功能，我们之前有个支付系统，就是用的Redis队列功能。

PubSub(发布订阅)是Redis2.0版本引入的消息传递模型。

顾名思义，消费者可以订阅一个或多个channel，生产者向对应channel发送消

息后，所有订阅者都能收到相关消息。对应channel发送消息后，所有订阅者都能收到相关消息。

在java代码中可以实现MessageListener接口，来消费队列中的消息。

```
@Slf4j
@Component
public class RedisMessageListenerListener implements MessageListener {
    @Autowired
    private RedisTemplate<String, Object> redisTemplate;

    @Override
    public void onMessage(Message message, byte[] pattern) {
        String channel = new String(pattern);
        RedisSerializer<?> valueSerializer = redisTemplate.getValueSerializer();
        Object deserialize = valueSerializer.deserialize(message.getBody());
        if (deserialize == null) return;
        String md5DigestAsHex = DigestUtils.md5DigestAsHex(deserialize.toString().getBytes(StandardCharsets.UTF_8));
        Boolean result = redisTemplate.opsForValue().setIfAbsent(md5DigestAsHex, "1", 20, TimeUnit.SECONDS);
        if (Boolean.TRUE.equals(result)) {
            log.info("接收的结果：{}", deserialize.toString());
        } else {
            log.info("其他服务处理中");
        }
    }
}
```

10. 生成全局ID

在有些需要生成全局ID的业务场景，其实也可以使用Redis。

可以使用incrby命令，利用原子性操作，可以执行下面这个命令：

```
incrby userid 10000
```

在分库分表的场景，对于有些批量操作，我们可以从Redis中，一次性拿一批id出来，然后给业务系统使用。

[8000页BAT大佬写的刷题笔记，让我offer拿到手软](http://cxyroad.com/”http://www.susan.net.cn/zt/8000.html”)

最后说一句(求，别白嫖我)

如果这篇文章对您有所帮助，或者有所启发的话，帮忙扫描下发二维码一下，您的支持是我坚持写作最大的动力。

求一键三连：点赞、转发、在看。

公众号：【苏三说技术】，在公众号中回复：面试、代码神器、开发手册、时间管理有超赞的粉丝福利，另外回复：加群，可以跟很多BAT大厂的前辈交流和学习。

原文链接: https://juejin.cn/post/7325132133168971813