

第36讲 | 如何制作游戏内容保存和缓存处理?

2018-08-16 蔡能

从0开始学游戏开发

[进入课程 >](#)



讲述：蔡能

时长 07:33 大小 3.47M



我们在打完游戏的关卡之后，需要保存游戏进度。单机游戏的进度都保存在本地磁盘上，如果是网络游戏的话该怎么办呢？这一节，我就来讲这个内容。

首先，我们要了解游戏内容的保存，需要先了解缓存处理。

为什么要了解缓存的处理呢？那是因为在大量用户的情况下，我们所保存的内容都是为了下次读取，如果每一次都从硬盘或者数据库读取，会导致用户量巨大数据库死锁，或者造成读取速度变慢，所以在服务器端，缓存的功能是一定要加上的。

Redis 不仅是内存缓存

缓存机制里有个叫 Redis 的软件。它是一种内存数据库，很多开发者把 Redis 当作单纯的内存缓存来使用，事实上，这种说法并不准确，Redis 完全可以当作一般数据库来使用。

Redis 是一种 key-value 型的存储系统。它支持存储的 value 类型很多，包括字符串、链表、集合、有序集合和哈希类型。这些数据类型都支持 push/pop、add/remove 及取交集并集和差集及更丰富的操作，而且这些操作都具有原子性。

Redis 还支持各种不同方式的排序。为了保证效率，数据一般都会缓存在内存中，而 Redis 会周期性地把更新的数据写入磁盘或者把修改操作写入追加的记录文件，并且在此基础上实现 master-slave（主从）的同步。

说到 Redis，就不得不说缓存机制的老前辈 Memcached。同样是缓存机制，Memcached 的做法是多线程，非阻塞的 IO 复用的网络模型。

多线程分监听线程和工作子线程。监听线程监听网络连接，接受请求了之后，将连接描述字使用管道传递给工作线程，进行读写。网络层的事件使用 libevent 封装。多线程模型可以发挥多核的作用。Memcached 所有操作都要对全局变量加锁，进行计数等工作，所以会有性能损耗。

而 Redis 使用单线程 IO 复用模型，自己封装了一个简单的事件处理框架，对于单纯只有 IO 操作的模型来说，单线程可以将速度优势发挥到最大，但是 Redis 也提供了一些简单的计算功能，比如排序、聚合等。

Redis 还可以在某些场景下对关系数据库（比如 MySQL）起到较好的补充作用。它提供了多种编程语言的接口，开发人员调用起来也很方便。


Redis 支持主从同步。通过配置文件，可以将主服务器上的数据往任意数量的从服务器上同步，从服务器 A1 也是主服务器 B（B 是关联到其他从服务器 B1，B2 的主服务器，同时又是主服务器 A 的从服务器 A1）。

这种做法就使得 Redis 可以执行单层的树结构的复制。Redis 实现了发布 / 订阅（publisher/subscriber）的机制。所谓发布和订阅，就是订阅者接收发布者的消息的时候，发布者和订阅者都不用去管对方是什么状态，只管各司其职就好了，在这种状态下，可以订阅一个频道并接收主服务器完整的消息发布记录。

编写 Redis 接口代码

我们尝试使用 Python 编写 Redis 接口的代码。


要使 Python 支持 Redis 编程，必须安装一个包“redis”，在使用的時候 import 一下。

 复制代码

```
1 import redis
```

然后我们开启 Redis 服务，在 Windows 下可以运行 redis-server.exe，使用默认配置即可。

现在，我们尝试使用代码连接一下数据库服务，并且往数据库存放并取出、删除内容。

 复制代码

```
1 r = redis.Redis(host='127.0.0.1', port=6379, db=0)
2 r.set('foo', 'my_redis')
3 print r.get('foo')
4 r.delete('foo')
5 print r.dbsize()
```


运行结果为输出 my_redis 和 0。

当然，如果我们没有运行 Redis，则会抛出一个异常：

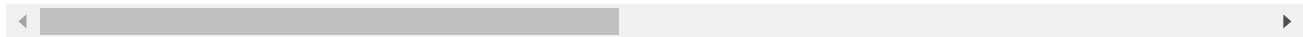
```
redis.exceptions.ConnectionError: Error 10061 connecting 127.0.0.1:6379. .
```

r 对象为连接 Redis 服务器的对象，其中 db=0 表示使用 redis 的 0 号数据库，可以随你喜欢切换为 1 号、2 号等等。如果 Redis 设置了密码，还可以在初始化的时候输入密码。

Redis 的初始函数是这样定义的：

 复制代码

```
1 __init__(self, host='localhost', port=6379, db=0, password=None, socket_timeout=None, co
```




在之后的代码中，`r.set` 表明将 key 为 `foo`，value 为 `my_redis` 的内容写入数据库。

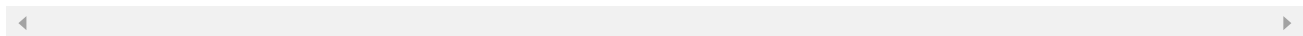
最后输出 0 号数据库的内容长度。

值得一提的是，Redis 对于存储的内容是来者不拒，有什么扔什么，所以你如果往 Redis 里插入二进制、UTF-8 编码、图片等等，任何东西都可以。理论上只要不超过内存大小的数据都可以往里面扔。

最后，我们可以这么写：

 复制代码

```
1 r.save()
```



强制 Redis 往硬盘里写入数据，这样我们就能保证数据不会因为电脑发生异常而丢失。这样就将内存的数据同步了下来。

我们常说的木桶理论其实在这里也适用。比如电脑的速度取决于电脑设备中最慢的那个设备，就像水在桶中的高度始终取决于水桶里面最下方的那个漏水处。而磁盘 I/O 始终是拖慢电脑速度的重要力量。

前面我们介绍了 Redis，所以我们可以使用 Redis 对文件进行缓存。Redis 可以当作普通缓存也可以当作文件缓存，在 Redis 中放入任何东西，当然也包括放入二进制文件，Redis 也不会有任何异常出现，从 Redis 缓存中取出二进制文件的速度也非常快，因为是直接从内存中取出数据。

我们假设网络游戏保存下来的数据很大，因为有人物属性、人物装备、地图 NPC 位置和怪兽等等。这些玩家退出后，游戏保存的数据文件，被保存在关系型数据库中，或者保存在服务器硬盘的文件中。我们不可能每次都去读取关系数据库中的游戏内容或者硬盘文件内容，所以，可以用一种方案来存放游戏保存的文件和缓存。

如何存放文件和缓存？

这套机制并不局限于读取保存文件，某些大文件，或者数据文件的读取和缓存上，都可以使用这种思路去做。

首先我们假定文件存放在某一个目录，所有的负载均衡服务器都存放有这个目录的副本，其他分布式服务器存放其他文件和目录，我们先暂定 A 服务器存放文件 A1、A2、A3。

这些都是游戏的保存文件，在服务器初始启动的时候，Redis 并不读取任何文件，当有请求过来的时候，服务器程序通知 Redis 读取某个文件。

这时，我们需要一个机制，为了保证服务器的内存开销，也为了保证缓存速度，我们必须保证被读取量最大的文件被缓存，而不是所有文件，这时候，Python 程序可以另开一个线程或者进程，暂且命名为 T 线程，记录某文件被缓存。

服务器程序每次得到请求的时候，都会将需要递交的被读取文件告诉 Python 线程 T，说文件 A1 被缓存了 N 次，文件 A2 被缓存了 N 次，在这种策略下，T 线程通过几个小时或者几天的计数，就能明确知道，比如 A2 文件被递交次数最多，于是它始终通知 Redis 将 A2 文件进行缓存，而 A1 由于到了某一天递交次数下降，在某一个时间节点上，线程 T 就告知 Redis A1 文件可以从缓存文件中撤出来，节省内存开销，让位给读取频次更高更高的文件。

这样，一套完整的缓存计数和缓存的解决方案就出现了。

当然，并不是说 MySQL 等关系型数据库不能做这些工作，但从效率和开发成本来讲，Redis（缓存）的开发成本和效率显然更胜一筹。因为在几十万几百万甚至上亿等级用户量的时候，就算是 Redis，在这种量级的情况下也是吃不消的，所以如果不在上层做更多层的缓存，底层数据库一定是会死锁或者出现各种各样的问题。

那么你可能会说，我可以做索引啊，要知道在连接数足够多的时候，做索引、读写分离，主从数据库等方案，也只是救急只用，无法真正实现稳固的架构体系。

小结

我来总结一下今天的内容。

Redis 不仅仅可以用作普通的缓存机制使用，也可以当作正常的数据库使用，Redis 也支持主从同步，要按照应用场景不同来配置不同的 Redis 使用场景。

缓存机制不仅仅针对读取游戏保存文件这么一种方案，也可以用作各种数据文件的读取和写入操作。

使用现成的 Redis 等缓存数据软件，是一个好的方案。而设计好的框架、好的缓存机制、好的网络模型，是一款好网游必不可少的条件。

现在给你留一个小问题吧。

有没有可能将网络游戏的内容保存在客户端本地的电脑上，如果可以的话，请问如果玩家换了一台电脑，怎么同步内容呢？保存在客户端本地的意义是什么？

欢迎留言说出你的看法。



从 0 开始学游戏开发

你的游戏开发入门第一课

蔡能 原网易游戏引擎架构师
资深游戏底层技术专家



新版升级：点击「 请朋友读」，10位好友免费读，邀请订阅更有**现金**奖励。

© 版权归极客邦科技所有，未经许可不得传播售卖。页面已增加防盗追踪，如有侵权极客邦将依法追究其法律责任。

上一篇 第35讲 | 如何用网关服务器进行负载均衡？

下一篇 课后阅读 | 游戏开发工程师学习路径（上）



风林

2018-08-16



大佬能不能加上对每次小问题的看法，我这个小白的很多问题没有头绪，想看看大佬的看法



放羊大王

2018-08-16



减压。做定时同步。

展开 ∨