Python中的垃圾回收机制

By 刘志军 (https://foofish.net/author/liu-zhi-jun.html), 2016-01-21, 分类： PYTHON技术 (https://foofish.net/category/pythonji-zhu.html)

gc (https://foofish.net/tag/gc.html), python (https://foofish.net/tag/python.html)

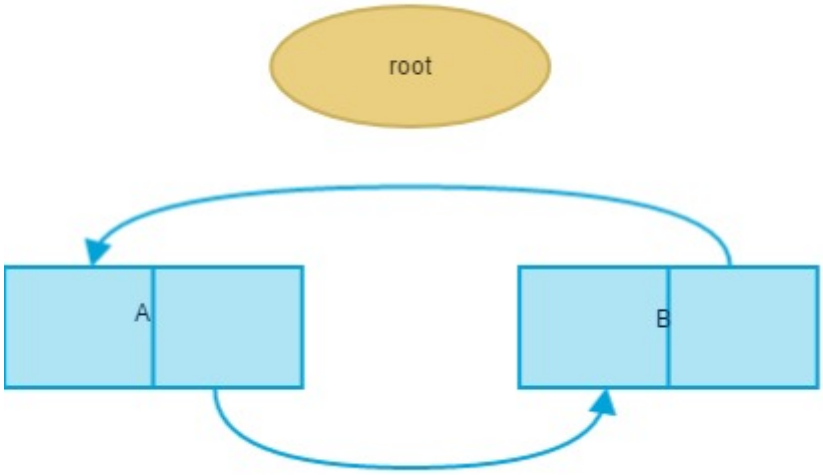
GC作为现代编程语言的自动内存管理机制，专注于两件事：1. 找到内存中无用的垃圾资源 2. 清除这些垃圾并把内存让出来给其他对象使用。GC彻底把程序员从资源管理的重担中解放出来，让他们有更多的时间放在业务逻辑上。但这并不意味着码农就可以不去了解GC，毕竟多了解GC知识还是有利于我们写出更健壮的代码。

引用计数

Python语言默认采用的垃圾收集机制是『引用计数法 Reference Counting』，该算法最早George E. Collins在1960的时候首次提出，50年后的今天，该算法依然被很多编程语言使用，『引用计数法』的原理是：每个对象维护一个 **ob_ref** 字段，用来记录该对象当前被引用的次数，每当新的引用指向该对象时，它的引用计数ob_ref加1，每当该对象的引用失效时计数ob_ref减1，一旦对象的引用计数为0，该对象立即被回收，对象占用的内存空间将被释放。它的缺点是需要额外的空间维护引用计数，这个问题是其次的，不过最主要的问题是它不能解决对象的"循环引用"，因此，也有很多语言比如Java并没有采用该算法做来垃圾的收集机制。

什么是循环引用？A和B相互引用而再没有外部引用A与B中的任何一个，它们的引用计数虽然都为1，但显然应该被回收，例子：

```
a = { } #对象A的引用计数为 1
b = { } #对象B的引用计数为 1
a['b'] = b  #B的引用计数增1
b['a'] = a  #A的引用计数增1
del a #A的引用减 1，最后A对象的引用为 1
del b #B的引用减 1，最后B对象的引用为 1
```

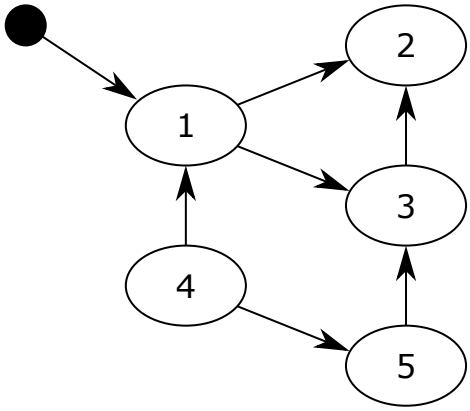


在这个例子中程序执行完 **del** 语句后，A、B对象已经没有任何引用指向这两个对象，但是这两个对象各包含一个对方对象的引用，虽然最后两个对象都无法通过其它变量来引用这两个对象了，这对GC来说就是两个非活动对象或者说是垃圾对象，但是他们的引用计数并没有减少到零。因此如果是使用引用计数法来管理这两对象的话，他们并不会被回收，它会一直驻留在内存中，就会造成了内存泄漏（内存空间在使用完毕后未释放）。为了解决对象的循环引用问题，Python引入了标记-清除和分代回收两种GC机制。

标记清除

『标记清除（Mark—Sweep）』算法是一种基于追踪回收（tracing GC）技术实现的垃圾回收算法。它分为两个阶段：第一阶段是标记阶段，GC会把所有的『活动对象』打上标记，第二阶段是把那些没有标记的对象『非活动对象』进行回收。那么GC又是如何判断哪些是活动对象哪些是非活动对象的呢？

对象之间通过引用（指针）连在一起，构成一个有向图，对象构成这个有向图的节点，而引用关系构成这个有向图的边。从根对象（root object）出发，沿着有向边遍历对象，可达的（reachable）对象标记为活动对象，不可达的对象就是要被清除的非活动对象。根对象就是全局变量、调用栈、寄存器。



在上图中，我们把小黑圈视为全局变量，也就是把它作为root object，从小黑圈出发，对象1可直达，那么它将被标记，对象2、3可间接到达也会被标记，而4和5不可达，那么1、2、3就是活动对象，4和5是非活动对象会被GC回收。

标记清除算法作为Python的辅助垃圾收集技术主要处理的是一些容器对象，比如list、dict、tuple，instance等，因为对于字符串、数值对象是不可能造成循环引用问题。Python使用一个双向链表将这些容器对象组织起来。不过，这种简单粗暴的标记清除算法也有明显的缺点：清除非活动的对象前它必须顺序扫描整个堆内存，哪怕只剩下小部分活动对象也要扫描所有对象。

分代回收

分代回收是一种以空间换时间的操作方式，Python将内存根据对象的存活时间划分为不同的集合，每个集合称为一个代，Python将内存分为了3“代”，分别为年轻代（第0代）、中年代（第1代）、老年代（第2代），他们对应的是3个链表，它们的垃圾收集频率与对象的存活时间的增大而减小。新创建的对象都会分配在年轻代，年轻代链表的总数达到上限时，Python垃圾收集机制就会被触发，把那些可以被回收的对象回收掉，而那些不会回收的对象就会被移到中年代去，依此类推，老年代中的对象是存活时间最久的对象，甚至是存活于整个系统的生命周期内。同时，分代回收是建立在标记清除技术基础之上。分代回收同样作为Python的辅助垃圾收集技术处理那些容器对象

参考：

- http://www.memorymanagement.org/mmref/recycle.html#tracing-collectors (http://www.memorymanagement.org/mmref/recycle.html#tracing-collectors)
- 《垃圾回收的算法与实现》
- 《Python源码剖析》

猜你喜欢

2017-07-01
Python是怎么火起来的 (https://foofish.net/why-is-python-so-popular.html)
2020-06-14
python合并两个列表 (https://foofish.net/python-merge-two-list.html)
2020-01-03
5个顶级异步Python框架 (https://foofish.net/python-async-web-framework.html)
2018-09-27
关于Python包和模块的10个知识清单 (https://foofish.net/package-module-tips.html)
2020-06-13
检查对象中是否存在某个属性 (https://foofish.net/python-check-attribute.html)
2014-02-17
Python多线程编程 (https://foofish.net/mutil-thread-programming.html)
2017-10-22
图解Python变量与赋值 (https://foofish.net/python-variable.html)
2019-05-13
python3判断list是否为空 (https://foofish.net/python-list-is-empty.html)
2020-06-07
求求你，别再手工造假数据了，fake了解一下 (https://foofish.net/python-fake-data.html)
2015-11-25
OS X El-Capitan 安装 virtualenvwrapper 遇到 Operation not permitted (https://foofish.net/six.html)

关于 (https://foofish.net/pages/about.html)

知乎 (https://www.zhihu.com/people/zhijunliu)

RSS (https://foofish.net/feeds/all.atom.xml)

Vimiix (https://www.vimiix.com/)

红色石头 (http://redstonewill.com/)

五分钟学算法 (https://www.cxyxiaowu.com/)

Python知识圈 (https://www.pyzhishiquan.com/)