

# 内存分配策略

作者: [xixiaoming](#)

原文链接: <https://ld246.com/article/1501234349697>

来源网站: [链滴](#)

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

# 对象优先在新生代的Eden上分配

## 大对象直接进入老年代

大对象就是需要大量连续内存的Java对象，如长字符串与数组，大对象可能导致内存还有不少剩余空间时就触发GC以获取足够的连续空间来安置这个大对象，所以写程序尽量避免大对象，特别是短命大对象

1) 为了避免在Eden和两个Survivor之间发生大量的内存拷贝，大对象直接在老年代分配（有个值可设置-XX:PretenureSizeThreshold，大于这个值就当作大对象直接到老年代去）

## 长期存活的对象将进入老年代

新生代的对象每熬过一次minor gc，年龄就增加1岁，当增加到一定程度（默认15岁）时，就会被晋到老年代

## 动态对象年龄判定

也并不是年龄达到15岁才能进入老年代的，如果在Survivor空间中相同年龄所有对象大小总和大于Survivor空间的一半，年龄大于或等于该年龄的对象就可以直接进入老年代

## 空间分配担保

在发生minor gc时，因为检测完会有一部分对象进入老年代，所以虚拟机先检测老年代的内存够不够（之前每次晋升到老年代的平均大小>老年代的剩余空间?），如果不够了，那么改为进行一次full gc，如果内存还够，那么检查是否允许担保失败，如果不允许，那么还是进行一次full gc（一般设置为允许担保失败以避免full gc过于频繁）