OS 课设7

信息科学与工程学院

2019011777 计算机 19-3 刘康来

实验 7 Web 服务器的内存分配

> 参考：`https://cloud.tencent.com/developer/article/1608933`,
> `https://blog.51cto.com/liangchaoxi/4050119`

- TCMalloc全称是Thread-Caching Malloc，是Google 开发的内存分配器，在不少项目中都有使用，例如在 Golang 中就使用了类似的算法进行内存分配。它具有现代化内存分配器的基本特征：对抗内存碎片、在多核处理器能够 scale。
- tcmalloc的内存分配分为四层：

1. ThreadCache（用于小对象分配）：线程本地缓存，每个线程独立维护一个该对象，多线程在并发申请内存时不会产生锁竞争。
2. CentralCache（Central free list，用于小对象分配）：全局cache，所有线程共享。当thread cache空闲链表为空时，会批量从CentralCache中申请内存；当thread cache总内存超过阈值，会进行内存垃圾回收，将空闲内存返还给CentralCache。
3. Page Heap（小/大对象）：全局页堆，所有线程共享。对于小对象，当centralcache为空时，会从page heap中申请一个span；当一个span完全空闲时，会将该span返还给page heap。对于大对象，直接从page heap中分配，用完直接返还给page heap。
4. 系统内存：当page cache内存用光后，会通过sbrk、mmap等系统调用向OS申请内存。

- 详细分配：

1. TCMalloc 会给每一个线程分配一个属于线程本地的缓存(Thread Cache)，这个本地缓存用于小对象(小于32K)的内存分配， 这样再多线程分配内存的情况下，可以减少锁竞争。

- 如果没有空闲内存，则从 Central Heap中一次性获取一连串小对象。从 Central Heap(备注：这个是多个线程分享的，操作的时候需要做加锁、解锁处理)移动到Thread Cache。

2. 对于大的对象(大于32K)则是直接从Central Heap按照页面层次分配方式进行内存分配。

- 使用页级分配器（page-level allocator）从Central page Heap中进行分配，一个大对象总是按页对齐的。

> tcmalloc对于小内存，按8的整数次倍分配，对于大内存，按4K的整数次倍分配。

3. 当某个线程缓存中所有对象的总大小超过2MB的时候，会进行垃圾收集。垃圾回收器(garbage collections)会周期性的将存储从Thread Cache(默认是2M)迁移到Central Heap

- 垃圾收集阈值会自动根据线程数量的增加而减少，这样就不会因为程序有大量线程而过度浪费内存。

- TCMalloc对于多线程程序而言，减少了锁机制，对于小对象而言，可以说没有锁的操作，对于大的对象而言的更加高效的自旋锁(spinlock)。
- TCMalloc对小对象的处理空间效率更好。

> 分配(不同对象处理) 和 释放内存(垃圾回收) 的效率高