# 内存管理器性能测试

## 一、性能测试

测试一：

系统：Mac OS 10.14.5

内存管理器：

- glibc malloc

- gperftools-2.7 tcmalloc

- jemalloc 5.2.0

大内存测试：依次分配(1~1024K)大小的内存，测试单个线程的平均运行时间。

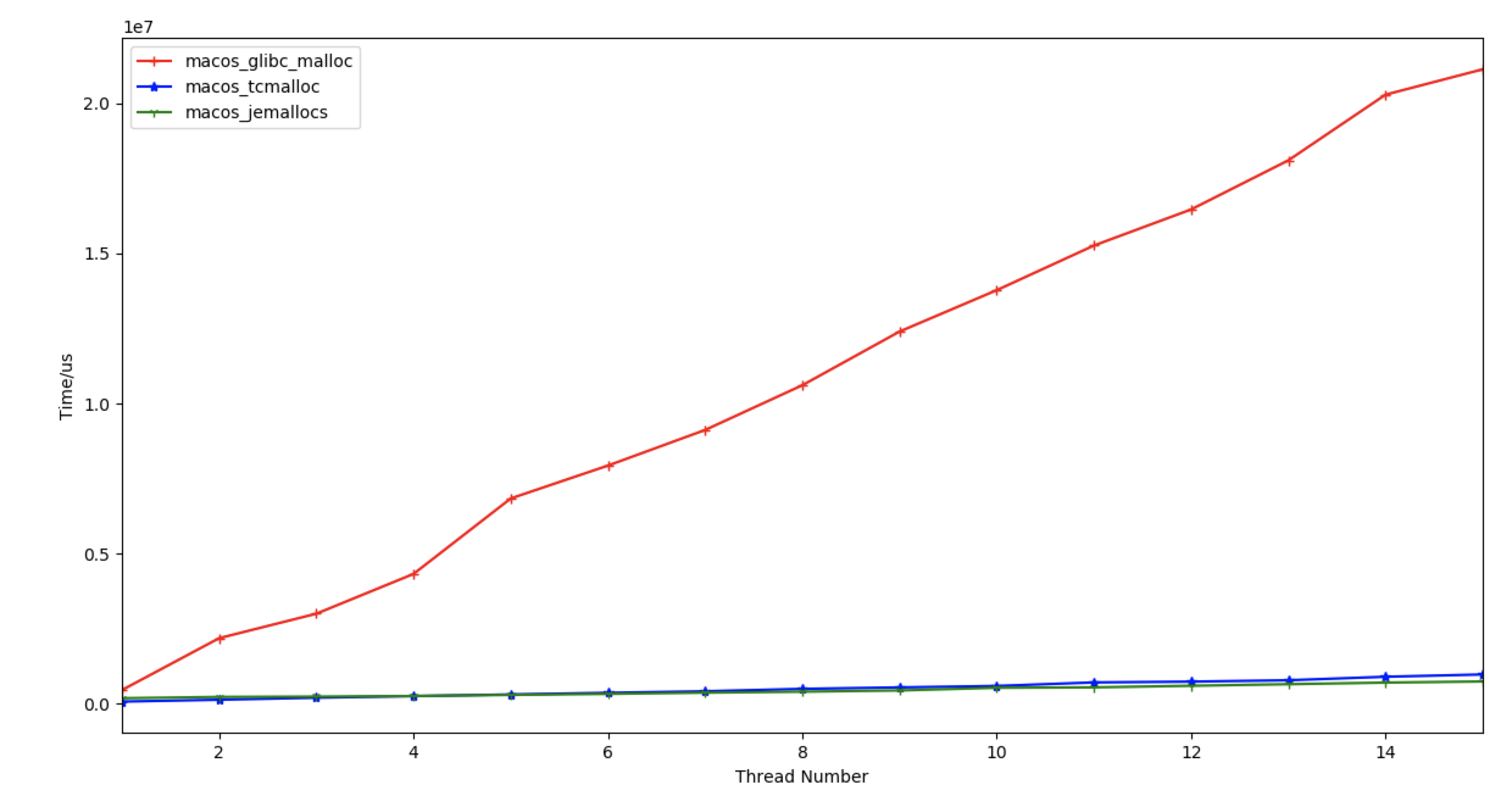


图 1 Mac OS 1~1024K内存管理测试

在MAC OS环境下，glibc自带的内存管理器，随着线程的增多，单个线程花费的时间线性增加，而使用tcmalloc以及jemalloc，则消耗的时间一直在某个阈值以下，且基本保持在缓速的增长中。

测试二：

系统：Mac OS 10.14.5

内存管理器：

- glibc malloc

- gperftools-2.7 tcmalloc

- jemalloc 5.2.0

小内存测试：依次分配(1~32K)大小的内存，测试单个线程的平均时间。

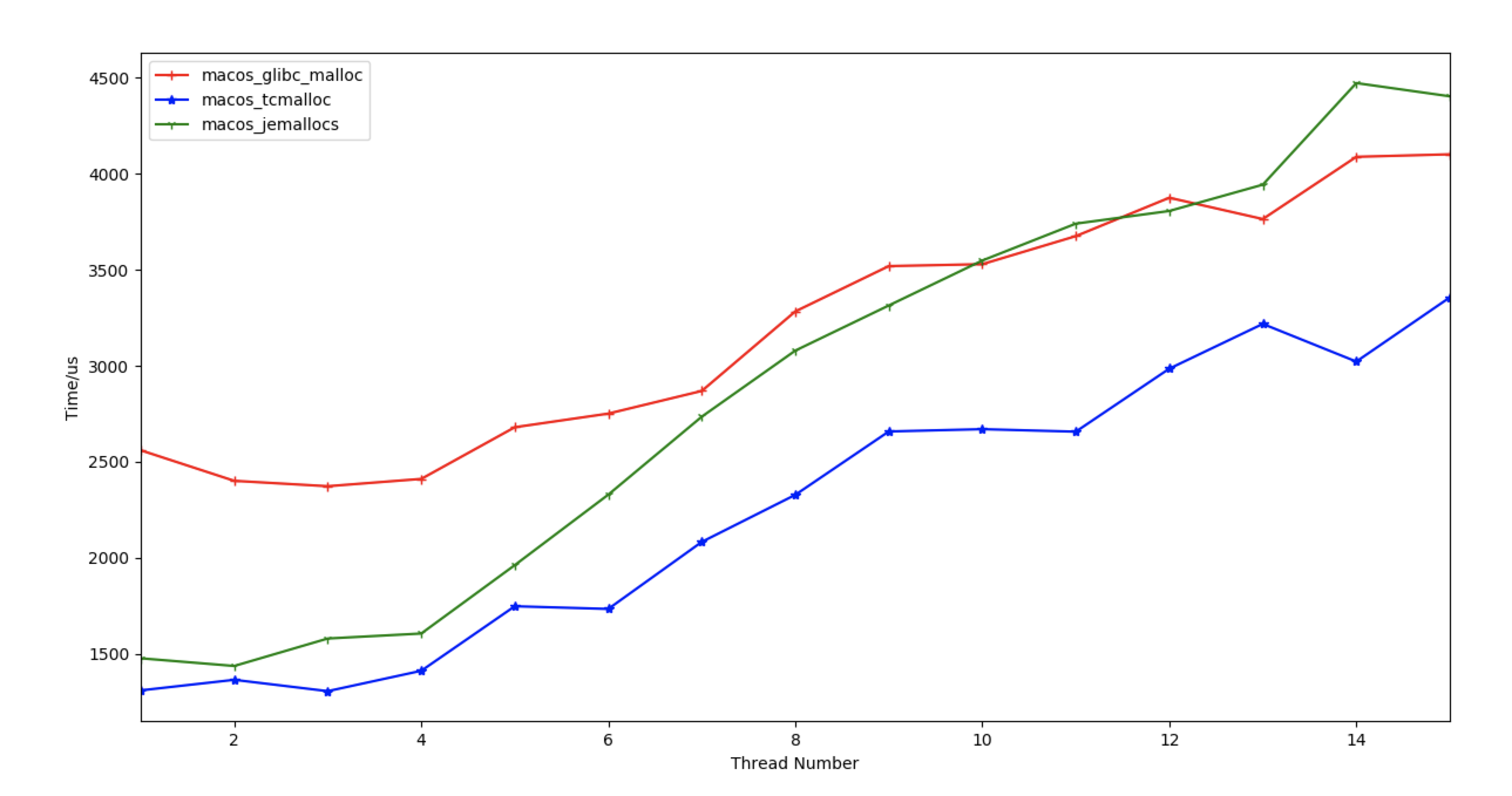


图 2 MAC OS (1~32K)内存管理测试

在小内存分配任务中，tcmalloc性能明显优于glibc malloc以及jemalloc，并且在线程数增多时，jemalloc的耗时，甚至超过了glibc malloc的耗时，由此可见，在Mac系统中，在小内存使用中，tcmalloc的性能更为优异。

测试三：

系统：Ubuntu 16.0

内存管理器：

- glibc malloc

- gperftools-2.7 tcmalloc

- jemalloc 5.2.0

大内存测试：依次分配(1~1024K)大小的内存，测试单个线程的平均运行时间。

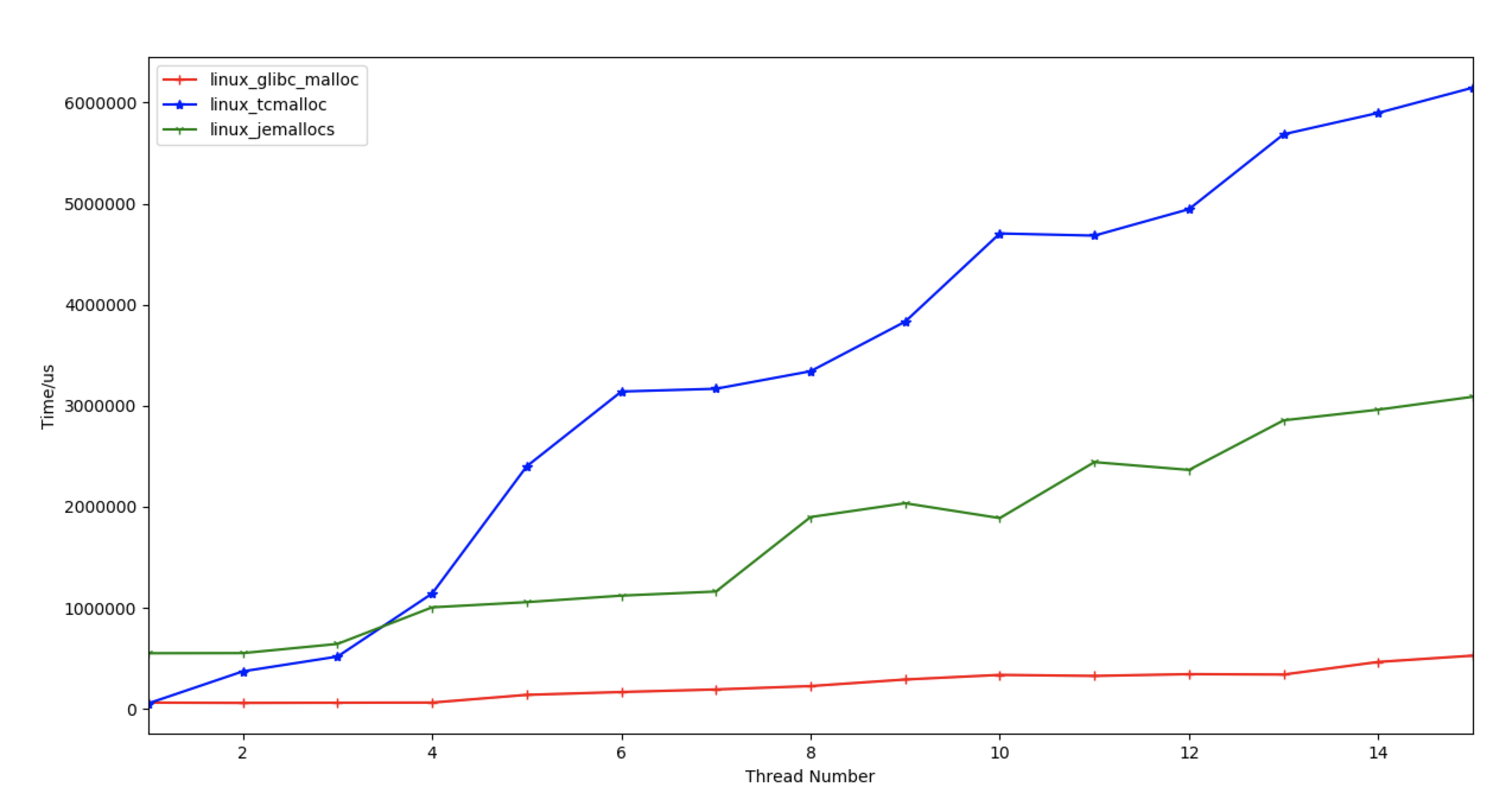


图 3 Ubuntu 16.0 (1~1024K)内存管理测试

在大内存的内存管理测试中，glibc malloc的性能反而优于tcmalloc以及jemalloc，这是因为tcmalloc以及jemalloc在申请大内存时，需要加锁，并且较大的时候，需要向Linux的memmap映射区申请内存，这些操作导致tcmalloc以及jemalloc的内存管理速度低于glibc malloc内存管理机制。而在大内存分配管理，tcmalloc又没jemalloc性能好。

测试四：

系统：Ubuntu 16.0

内存管理器：

- glibc malloc

- gperftools-2.7 tcmalloc

- jemalloc 5.2.0

小内存测试：依次分配(1~32)大小的内存，测试单个线程的平均运行时间。

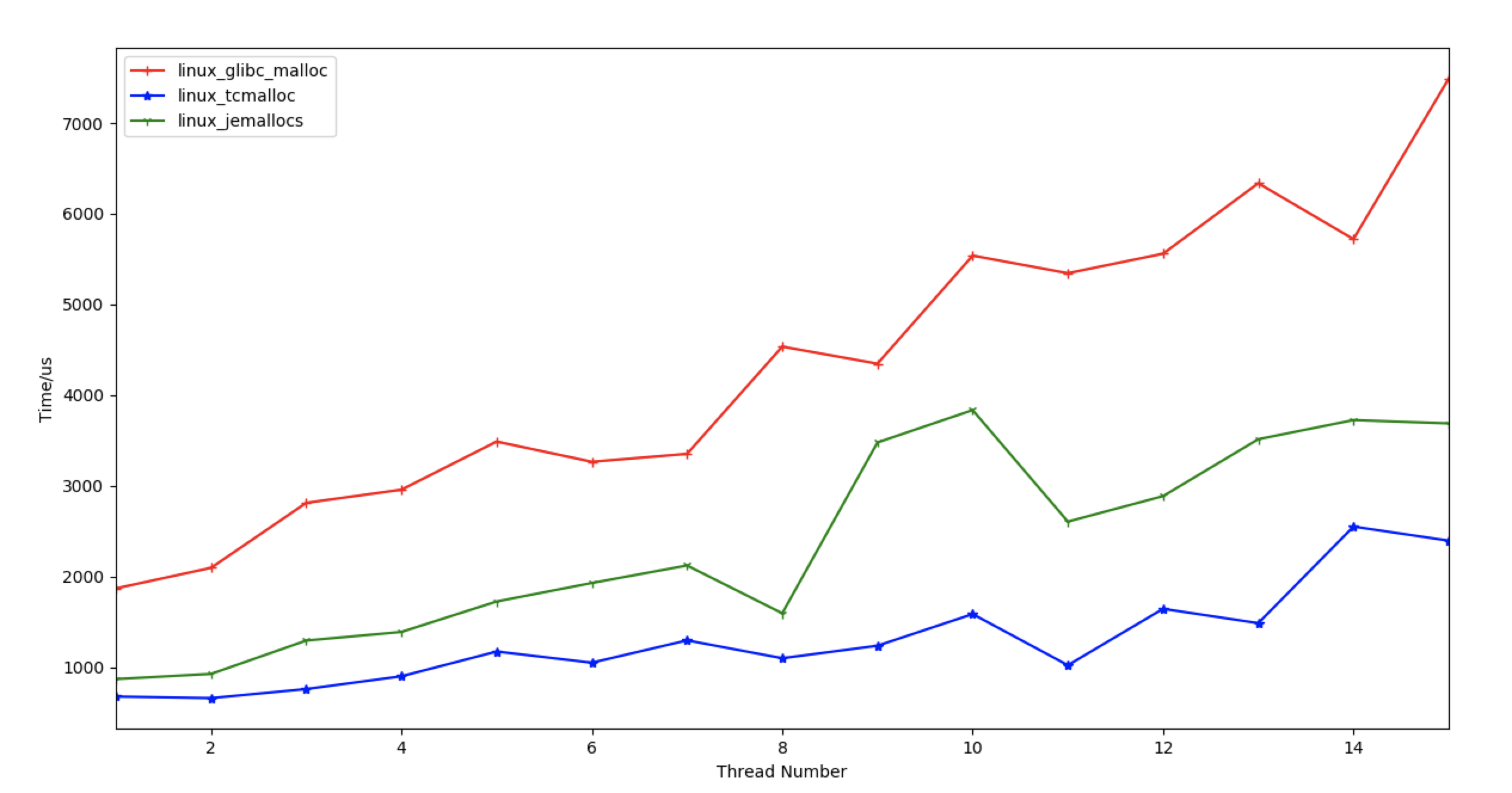


图 4 Ubuntu 16.0 (1~32K)内存管理测试

在Linux系统的小内存分配管理测试中，其效果与MAC OS测试结果相同，tcmalloc的性能最好，jemalloc的性能次之，glibc malloc性能最差。tcmalloc以及jemalloc随着线程数的增加，其时间消耗增长缓慢。

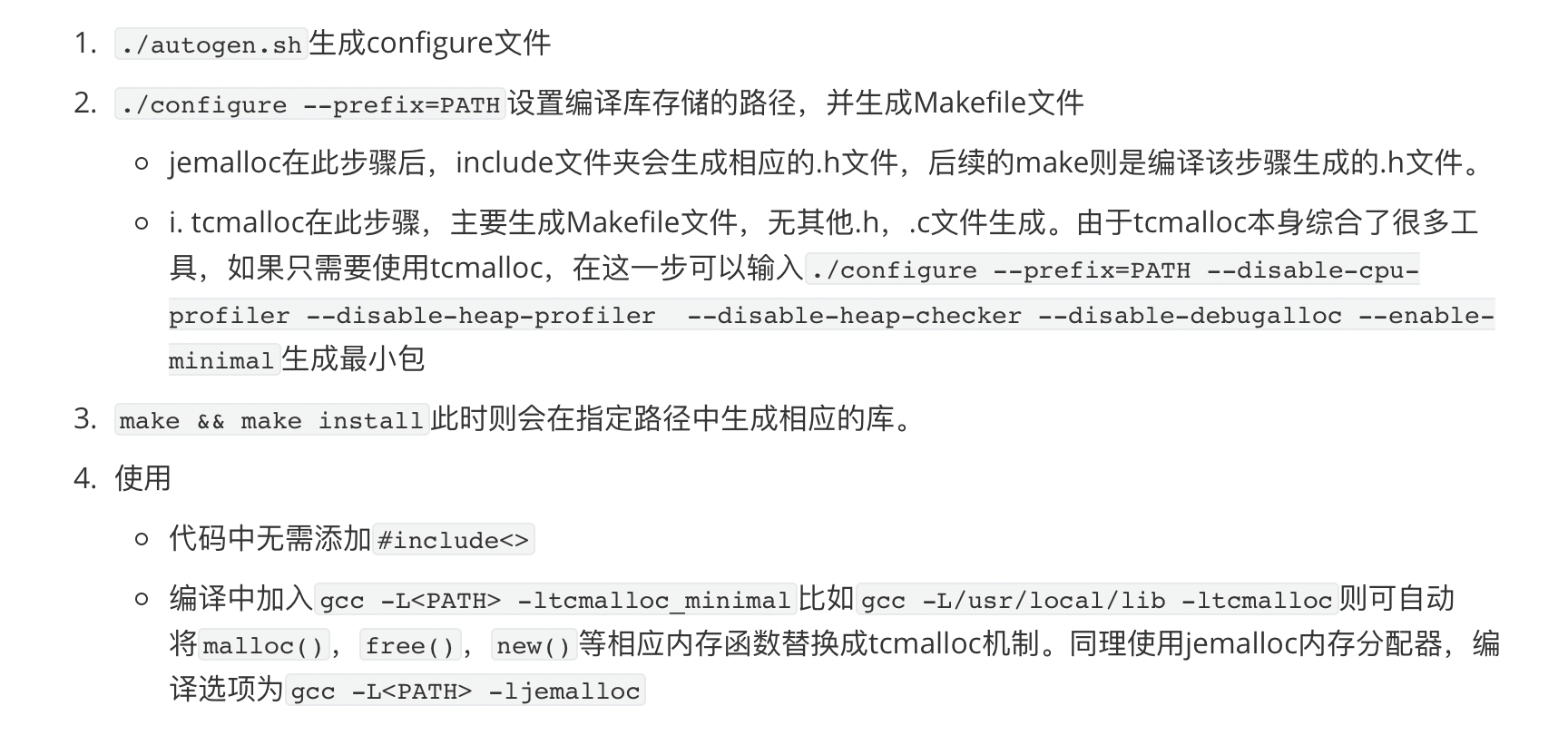
**结论：在大内存分配时，不同系统由于其系统低层机制的不同，不同的内存管理器具有不同的性能表现。但是在小内存时(<32K)，无论是tcmalloc，亦或是jemalloc其性能都高于glibc malloc(ptmalloc)，且tcmalloc较jemalloc，具备更好的性能，远优于glibc malloc。**

## tcmalloc与jemalloc的编译与使用

预安装：

autoconf, autogen等程序包。

MAC OS：



Linux:

