

谈一谈至关重要的Linux系统Swap交换区 PDF转换可能丢失图片或格式，建议阅读原文

https://www.100test.com/kao_ti2020/234/2021_2022__E8_B0_88_E4_B8_80_E8_B0_88_E8_c67_234022.htm Swap，即交换区，除了安装Linux的时候，有多少人关心过它呢？其实，Swap的调整对Linux服务器，特别是Web服务器的性能至关重要。通过调整Swap，有时可以越过系统性能瓶颈，节省系统升级费用。Swap的原理是一个较复杂的问题，需要大量的篇幅来说明。在这里只作简单的介绍，在以后的文章中将与大家详细讨论Swap实现的细节。众所周知，现代操作系统都实现了“虚拟内存”这一技术，不但在功能上突破了物理内存的限制，使程序可以操纵大于实际物理内存的空间，更重要的是，“虚拟内存”是隔离每个进程的安全保护网，使每个进程都不受其它程序的干扰。Swap空间的作用可简单描述为：当系统的物理内存不够用的时候，就需要将物理内存中的一部分空间释放出来，以供当前运行的程序使用。那些被释放的空间可能来自一些很长时间没有什么操作的程序，这些被释放的空间被临时保存到Swap空间中，等到那些程序要运行时，再从Swap中恢复保存的数据到内存中。这样，系统总是在物理内存不够时，才进行Swap交换。计算机用户会经常遇这种现象。例如，在使用Windows系统时，可以同时运行多个程序，当你切换到一个很长时间没有理会的程序时，会听到硬盘“哗哗”直响。这是因为这个程序的内存被那些频繁运行的程序给“偷走”了，放到了Swap区中。因此，一旦此程序被放置到前端，它就会从Swap区取回自己的数据，将其放进内存，然后接着运行。需要说明一点，并不是所有从物理内存

中交换出来的数据都会被放到Swap中（如果这样的话，Swap就会不堪重负），有相当一部分数据被直接交换到文件系统。例如，有的程序会打开一些文件，对文件进行读写（其实每个程序都至少要打开一个文件，那就是运行程序本身），当需要将这些程序的内存空间交换出去时，就没有必要将文件部分的数据放到Swap空间中了，而可以直接将其放到文件里去。如果是读文件操作，那么内存数据被直接释放，不需要交换出来，因为下次需要时，可直接从文件系统恢复；如果是写文件，只需要将变化的数据保存到文件中，以便恢复。但是那些用malloc和new函数生成的对象的数据则不同，它们需要Swap空间，因为它们在文件系统中没有相应的“储备”文件，因此被称作“匿名” (Anonymous)内存数据。这类数据还包括堆栈中的一些状态和变量数据等。所以说，Swap空间是“匿名”数据的交换空间。

突破128M Swap限制

经常看到有些Linux（国内汉化版）安装手册上有这样的说明：Swap空间不能超过128M。为什么会有这种说法？在说明“128M”这个数字的来历之前，先给问题一个回答：现在根本不存在128M的限制！现在的限制是2G！Swap空间是分页的，每一页的大小和内存页的大小一样，方便Swap空间和内存之间的数据交换。旧版本的Linux实现Swap空间时，用Swap空间的第一页作为所有Swap空间页的一个“位映射”（Bit map）。这就是说第一页的每一位，都对应着一页Swap空间。如果这一位是1，表示此页Swap可用；如果是0，表示此页是坏块，不能使用。这么说来，第一个Swap映射位应该是0，因为，第一页Swap是映射页。另外，最后10个映射位也被占用，用来表示Swap的版本（原来的版本是Swap_space，现在的版本

是swapspace2)。那么，如果说一页的大小为s，这种Swap的实现方法共能管理“ $8 * (s - 10) - 1$ ”个Swap页。对于i386系统来说s=4096，则空间大小共为133890048，如果认为1 MB=2²⁰ Byte的话，大小正好为128M。之所以这样来实现Swap空间的管理，是要防止Swap空间中有坏块。如果系统检查到Swap中有坏块，则在相应的位映射上标记上0，表示此页不可用。这样在使用Swap时，不至于用到坏块，而使系统产生错误。现在的系统设计者认为：1.现在硬盘质量很好，坏块很少。2.就算有，也不多，只需要将坏块罗列出来，而不需要为每一页建立映射。3.如果有很多坏块，就不应该将此硬盘作为Swap空间使用。于是，现在的Linux取消了位映射的方法，也就取消了128M的限制。直接用地址访问，限制为2G。

Swap配置对性能的影响 分配太多的Swap空间会浪费磁盘空间，而Swap空间太少，则系统会发生错误。如果系统的物理内存用光了，系统就会跑得很慢，但仍能运行；如果Swap空间用光了，那么系统就会发生错误。例如，Web服务器能根据不同的请求数量衍生出多个服务进程（或线程），如果Swap空间用完，则服务进程无法启动，通常会出现“application is out of memory”的错误，严重时会造成服务进程的死锁。因此Swap空间的分配是很重要的。通常情况下，Swap空间应大于或等于物理内存的大小，最小不应小于64M，通常Swap空间的大小应是物理内存的2-2.5倍。但根据不同的应用，应有不同的配置：如果是小的桌面系统，则只需要较小的Swap空间，而大的服务器系统则视情况不同需要不同大小的Swap空间。特别是数据库服务器和Web服务器，随着访问量的增加，对Swap空间的要求也会增加，具体配

置参见各服务器产品的说明。另外，Swap分区数量对性能也有很大的影响。因为Swap交换的操作是磁盘IO的操作，如果有多个Swap交换区，Swap空间的分配会以轮流的方式操作于所有的Swap，这样会大大均衡IO的负载，加快Swap交换的速度。如果只有一个交换区，所有的交换操作会使交换区变得很忙，使系统大多数时间处于等待状态，效率很低。用性能监视工具就会发现，此时的CPU并不很忙，而系统却慢。这说明，瓶颈在IO上，依靠提高CPU的速度是解决不了问题的。[NextPage]系统性能监视 Swap空间的分配固然很重要，而系统运行时的性能监控却更加有价值。通过性能监视工具，可以检查系统的各项性能指标，找到系统性能的瓶颈。本文只介绍一下在Solaris下和Swap相关的一些命令和用途。最常用的是Vmstat命令（在大多数Unix平台下都有这样一些命令），此命令可以查看大多数性能指标。例如：

```
# vmstat 3
procs -----memory----- -swap- --io-- -system- ---cpu--- r b
wswpd free buff cache si so bi bo in cs us sy id 0 0 0 0 93880
330419372 0 0 10 2 131 10 0 0 99 0 0 0 0 93880 3304 19372 0 0 0 0109
8 0 0 100 0 0 0 0 93880 3304 19372 0 0 0 0 112 6 0 0 100 .....
```

100Test 下载频道开通，各类考试题目直接下载。详细请访问
www.100test.com