



链滴

Java 性能调优工具箱之操作系统的工具和分析 (二)

作者: [AutisticV5](#)

原文链接: <https://ld246.com/article/1573455017393>

来源网站: [链滴](#)

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

1. 磁盘使用率

监控磁盘使用率有两个目的：第一个目的与应用有关。如果应用正在做大量的磁盘I/O操作，那I/O就容易称为瓶颈；第二个目的是即使预计应用不会有很高的I/O，有助于监控系统是否在进行内存交换。

想了解何时磁盘I/O是瓶颈非常困难，因为这取决于应用的行为。如果应用往磁盘写数据时，没有有的缓冲，磁盘I/O的统计数据就会非常低。但是，如果应用执行的I/O超过了磁盘的承载，磁盘I/O的计数据就会非常高。这两种情形的性能都需要提升。

这是Linux系统iostat的部分输出：

```
[root@iZ2ze8dv3a3meruy2z6bkgZ ~]# iostat -xm 5
Linux 3.10.0-1062.1.2.el7.x86_64 (iZ2ze8dv3a3meruy2z6bkgZ)      11/11/2019      _x86_64_      (2 CPU)

avg-cpu:  %user   %nice %system %iowait  %steal   %idle
           0.28    0.00   0.21   0.02   0.00   99.49

Device:            rrqm/s   wrqm/s     r/s     w/s    rMB/s    wMB/s avgrq-sz avgqu-sz   await  r_await  w_await  svctm  %util
vda                  0.00     0.31    0.11    0.46     0.00     0.01   38.26     0.01   15.17    1.42   18.48   0.76   0.04

avg-cpu:  %user   %nice %system %iowait  %steal   %idle
           0.30    0.00   0.20   0.00   0.00   99.50

Device:            rrqm/s   wrqm/s     r/s     w/s    rMB/s    wMB/s avgrq-sz avgqu-sz   await  r_await  w_await  svctm  %util
vda                  0.00     0.00     0.00     0.00     0.00     0.00    0.00     0.00    0.00    0.00    0.00   0.00   0.00

avg-cpu:  %user   %nice %system %iowait  %steal   %idle
           0.20    0.00   0.10   0.00   0.00   99.70

Device:            rrqm/s   wrqm/s     r/s     w/s    rMB/s    wMB/s avgrq-sz avgqu-sz   await  r_await  w_await  svctm  %util
vda                  0.00     0.00     0.00     0.00     0.00     0.00    0.00     0.00    0.00    0.00    0.00   0.00   0.00
```

avg-cpu字段说明：

字段名	字段名解释
%user	CPU处在用户模式下的时间百分比
%nice 比	CPU处在带NICE值的用户模式下的时间百分比
%system	CPU处在系统模式下的时间百分比
%iowait	CPU等待输入输出完成时间的百分比
%steal PU的无意识等待时间百分比	管理程序维护另一个虚拟处理器时，虚拟PU的无意识等待时间百分比
%idle	CPU空闲时间百分比

device字段说明：

字段名	字段名解释
rrqm/s	每秒这个设备相关的读取请求有多少被Merge了(当系统调用需要读取数据的时候，VFS将请求发到各个FS，如果FS发现不同的读取请求读取的相同Block的数据，FS会将这个请求合并Merge)
wrqm/s	每秒这个设备相关的写入请求有多少被Merge了
r/s	每秒向磁盘发起的读操作数
w/s	每秒向磁盘发起的写操作数
rMB/s	每秒读M字节数
wMB/s	每秒写M字节数
avgrq-sz	平均每次设备I/O操作的数据大小

avgqu-sz 平均请求队列的长度，队列长度越短越好。

await 平均每次设备I/O操作的等待时间(毫秒)，一般地，系统I/O响应时间应该低于5ms，如果大于10ms就比较大了

r_await 每个读操作平均所需的时间；不仅包括盘设备读操作的时间，还包括了在kernel队列中等待的时间

w_await 每个写操作平均所需的时间；不仅包括盘设备写操作的时间，还包括了在kernel队列中等待的时间

svctm 平均每次设备I/O操作的服务时间(毫秒)(个数据不可信)

%util 一秒中有百分之多少的时间用于I/O操作，被IO消耗的CPU百分比，一般地，如果该参数是100%表示设备已经接近满负荷运行了

性能监控指标：

- **%iowait**：如果该值较高，表示磁盘存在I/O瓶颈
- **await**：一般地，系统I/O响应时间应该低于5ms，如果大于10ms就比较大了
- **avgqu-sz**：如果I/O请求压力持续超出磁盘处理能力，该值将增加。如果单块磁盘的队列长度超过2，一般认为该磁盘存在I/O性能问题。需要注意的是，如果该磁盘为磁盘阵列虚拟的逻辑驱动器需要再将该值除以组成这个逻辑驱动器的实际物理磁盘数目，以获得平均单块硬盘的I/O等待队列长度
- **%util**：一般地，如果该参数是100%表示设备已经接近满负荷运行了

小结

- 对于所有应用来说，监控磁盘使用率非常重要。即便不直接写磁盘的应用，系统交换仍然会影响它们的性能。
- 写入磁盘的应用遇到瓶颈，是因为写入数据的效率不高(吞吐量太低)，或者是因为写入太多数据(吞吐量太高)

2. 网络使用率

如果应用运行时需要网络，比如Java EE应用服务器，你也必须监控网络流量。网络使用率类似磁盘量。应用可能没有充分利用网络所以带宽太低，或者写入某网络接口的总数据量超过了它所能处理得。

有许多开源和商业工具可以监控网络带宽。Unix里一个受欢迎的命令行工具就是nicstat，它可以显示个网络接口的流量概要，包括接口的使用度。

nicstat工具的安装

下载地址：<https://sourceforge.net/projects/nicstat/>

下载后安装

```
tar -zxvf nicstat-1.95.tar.gz
```

```
cd nicstat-1.95/
```

```
cp Makefile.Linux Makefile
```

注意：Makefile里面默认是用32位系统下编译的，需要修改成64位。

```
vi Makefile
```

将CFLAGS = (COPT) (CMODEL) 修改为 CFLAGS = \$(COPT)

```
make
```

make install

这是Linux系统nicstat的输出

```
[root@iZ2ze8dv3a3meruy2z6bkgZ nicstat-1.95]# nicstat 5
Time      Int      rKB/s    wKB/s    rPk/s    wPk/s    rAvs    wAvs %Util    Sat
09:51:30 vethd496a6d  0.14    0.02    0.15    0.17    967.3    140.4  0.00    0.00
09:51:30   eth0    2.01    0.69    1.81    1.21   1142.0    585.5  0.00    0.00
09:51:30     lo    0.16    0.16    0.30    0.30    541.2    541.2  0.00    0.00
09:51:30 veth001c4cf  0.31    0.28    0.26    0.23   1204.1   1237.5  0.00    0.00
09:51:30 docker0  0.44    0.30    0.42    0.41   1095.9    769.5  0.00    0.00
Time      Int      rKB/s    wKB/s    rPk/s    wPk/s    rAvs    wAvs %Util    Sat
09:51:35 vethd496a6d  0.00    0.00    0.00    0.00    0.00    0.00  0.00    0.00
09:51:35   eth0    0.01    0.02    0.20    0.20    60.00    106.0  0.00    0.00
09:51:35     lo    0.00    0.00    0.00    0.00    0.00    0.00  0.00    0.00
09:51:35 veth001c4cf  0.00    0.00    0.00    0.00    0.00    0.00  0.00    0.00
09:51:35 docker0  0.00    0.00    0.00    0.00    0.00    0.00  0.00    0.00
```

字段名

字段名解释

Int

网络接口设备名

rKB/s

每秒读取kb数

wKB/s

每秒写入kb数

rPk/s

每秒读取的包数

wPk/s

每秒写入的包数

rAvs

每次读取平均字节

wAvs

每次写入平均字节

%Util

网络接口使用率

Sat

饱和度

在做性能压测的时候，最关键看网络接口使用率和饱和度。

小结：

1. 对基于网络的应用来说，务必要监控网络以确保它不是瓶颈
2. 往网络写数据的应用遇到瓶颈，可能是因为写数据的效率太低(吞吐量太低)，也可能是因为写入太多数据(吞吐量太高)