Why is my system so slow?

This is for Linux and Unix systems. Search engines find it for Windows people looking for info about the "AVSERVE.EXE" process, but this isn't going to help you with that. You are probably infected by Sasser or some other worm/virus.

This is not a performance tuning article. If your system is always slow, this article may not be what you are looking for. I'll be covering some general performance related issues here, but the main focus is for the system that was running fine yesterday but is sucking mud today. The typical response to such problems is "Reboot it", and while that may indeed fix the problem, it does not address the root cause, so you are likely to have the situation again.

You may also want to review General Troubleshooting

You need to figure out WHY it is slow. It would be nice if you had prepared ahead of time by having such tools as "sar" enabled for historical data or at least were familiar with what "top" and "vmstat" look like when your system is "normal"- that is, doing whatever tasks it is supposed to do, with the usual number of users, and not seeming "slow". A baseline of "normal" performance is very useful when trying to figure out what is causing abnormal performance. For instance, here's a "sar -b" from a Linux system:

Linux 2.4.9-12 (apl) 12/03/2001

07:01:00 AM	tps	rtps	wtps	bread/s	bwrtn/s
07:11:00 AM	1.28	0.00	1.28	0.03	11.93
07:21:00 AM	1.34	0.10	1.25	2.53	12.37
08:01:00 AM	1.00	1.00	1.00	1.00	1.00
08:11:00 AM	32.51	28.41	4.10	235.71	64.94
08:21:00 AM	111.83	101.62	10.21	7127.43	181.95
08:31:00 AM	75.79	52.29	23.50	2348.18	625.73
Average:	400.01	335.11	64.91	20863.12	1691.67

Does the activity beginning just after 8:00 AM represent an abnormal condition for this machine? Will the system seem slow with this amount of disk activity? If the system is slow right now, is this why- is it because of disk access? Suppose I now take a "sar -b 5 10" sample and see this:

Linux 2.4.9-12 (apl) 12/03/2001

08:32:20 AM	tps	rtps	wtps	bread/s	bwrtn/s
08:32:25 AM	0.00	0.00	0.00	0.00	0.00
08:32:30 AM	0.00	0.00	0.00	0.00	0.00
08:32:35 AM	0.00	0.00	0.00	0.00	0.00
08:32:40 AM	158.80	0.40	158.40	3.20	5360.40
08:32:45 AM	143.60	3.60	140.00	67.20	2299.60
08:32:50 AM	45.20	44.60	0.60	356.80	6.40
08:32:55 AM	0.60	0.00	0.60	0.00	6.40
08:33:00 AM	2.80	0.00	2.80	0.00	32.00
08:33:05 AM	32.00	20.60	11.40	164.80	1225.20
08:33:10 AM	43.20	43.00	0.20	344.00	102.40
Average:	42.62	11.22	31.40	93.60	903.24

What does that indicate about the disk activity?

By the way, the tools you have available will vary. For example, "sar" on SCO Unix is far more comprehensive than it is on Linux. On the other hand, Linux exposes a wealth of information through the /proc filesystem that is much more difficult to obtain on SCO.

That's no longer true. The linux sar in the "sysstat" package is quite complete now.

Well, in fact, this wouldn't be a particularly heavy load on the hardware I ran it on, and wouldn't noticeably affect performance on the single user desktop that it is. The "sar -b 5 10" shows that whatever the disk activity is, it isn't constant. But is it "normal"? The only way you'd know that is if your were familiar with the system or could look at historical data- which is why it is important to collect that data from the very first day you put a machine into service.

Sar

Do you have the proper patches on your system?
With ANY problem, there really is no sense in chasing it very far if you are running on systems that are not properly patched. Updated programs and kernel fixes often make problems magicaly go away.

Your beginning tool for that is "sar"- enable it on SCO Unix if it isn't already enabled with

/usr/lib/sa/sar_enable -y

Ignore the warning about rebooting- it is not necessary.

If you already know about sar, you may want to skip ahead to What's Wrong? now.

All that does is uncomment entries for sa1 and sa2 in the sys and root crontabs. By default, SCO runs sar every twenty minutes during "working hours" and ever hour otherwise; you should adjust the "sys" crontab to meet your specific needs. The daily summary (the sa2 script run from root's crontab) creates summary files in /var/adm/sa. These will have names like "sar01", "sar02", etc.; the daily data files are named "sa01" through "sa31". The daily data files are binary data. If you wanted to examine memory statistics from the 15th of the month, you'd run

sar -f /var/adm/sa15 -r

The "sar" summary files are text. The "sar15" is the output of running

sar -f /var/adm/sa15 -A > /var/adm/sar15

and therefor can be viewed directly with "more", printed directly or whatever.

On Red Hat Linux systems, sar data collection runs from /etc/cron.hourly/sysstat:

#!/bin/sh

```
# snapshot system usage every 10 minutes six times.
umask 0022
/usr/lib/sa/sa1 600 6 &
```

The daily summary is done by:

```
#!/bin/sh

# generate a daily summary of process accounting.
umask 0022
/usr/lib/sa/sa2 -A &
```

The sar files are found in /var/log/sa, and use the same naming as on SCO. If you have been running sar for a month or more, you'll always have one months worth of historical data. As daily files are overwritten by daily files from the following month, you don't have to be concerned with using up disk space. Having this historical data lets you quickly decide if the current sar statistics represent an unusual condition.

The flags for sar vary on different OS's, so read the man page. On all systems, sar without any arguments gives you cpu usage, but even there the output will vary. Linux systems have a "nice" column that SCO Unix lacks, and SCO includes a useful "wio" column not found on Linux:

```
Linux 2.4.9-12 (apl)    12/04/2001

05:32:15 AM       CPU     %user     %nice   %system     %idle
05:32:20 AM       all      0.00      0.00      0.20     99.80
05:32:25 AM       all      0.00      0.00      0.00    100.00
05:32:30 AM       all      4.20      0.00      1.00     94.80
05:32:35 AM       all      4.40      0.00      0.60     95.00
05:32:40 AM       all      0.00      0.00      0.00    100.00
Average:          all      1.72      0.00      0.36     97.92


SCO_SV scosysv 3.2v5.0.6 PentIII    12/04/2001

05:48:33    %usr    %sys    %wio   %idle (-u)
05:48:38       0       0       0     100
05:48:43       0       0       0     100
05:48:48       0       0       0     100
05:48:53       0       0       0     100
05:48:58       0       0       0     100

Average        0       0       0     100
```

If you run sar without any numerical arguments, it will look for today's historical data (and complain if it can't find it). If you run it with numerical arguments, it samples what is happening now. The first argument is the time between samples (5 seconds is a good choice), the second is the number of samples. So "sar 5 2" gives two samples, 5 seconds apart.

What's Wrong?

So, the system is slow. Let's try to find out why.

If it is the cpu that is pegged busy, it *may* be a run away process that is eating cpu cycles. Do this:

```
for x in 1 2 3 4 5
do
ps -e | sort -r +2 | head -5
echo "==="
sleep 5
```

```
done
```

Look for a process who's time column has gone up by 3 to 5 seconds each time- if you have something like that, that's your problem- you need to kill it. The TIME column is time on the cpu- normally a process doesn't spend a great deal of time actually running- it's waiting for the disk, waiting for you to type something, etc. Most processes spend most of their time sleeping, waiting for something else to happen, so something that gains 3 seconds or more in 5 seconds of wall time is usually suspect.

If you watch it over a few minutes, the time it gains here divided by the elapsed wall clock time is the percentage of your cpu this process is taking for itself. A shortlived process can take a lot of the cpu to print, or to redraw an X screen etc., so you have to use some good judgement here. But 3 seconds out of 5 is very likely a real problem.

Of course you need to understand what you are killing: you probably wouldn't want to kill the main Oracle database, for example.

If you kill the errant process and another copy of it pops right back to the top of the list, then you need to track down its parent:

```
# for example, if process 15246 is the problem
ps -p 15246 -o ppid
```

Of course, it may go further up the chain. Here's a script that traces back to init:

```
# This works on SCO or Linux, just pass a process ID as an argument.
MYPROC=$1
NEXTPROC=$MYPROC
while [ $NEXTPROC != 0 ]
do
ps -lp $NEXTPROC
MYPROC=$NEXTPROC
NEXTPROC=`ps -p $MYPROC -o "ppid=" `
done
```

Sometimes you'll have a badly written network program that starts sucking resources when its client dies. If you can't get the supplier to fix it, you may want to write a script to track down and kill these things. One clue that might help: the difference between a good "xyz" process and a bad one might just be whether or not it has an attached tty. So, if you see this:

```
  5821       ?    00:00:42 xyz
6689   ttyp0    00:00:08 xyz
7654   ttyp1    00:00:12 xyz
```

It's probably the one with a "?" that will start accumulating time. So a script that watched for and killed those might look like this:

```
set -f
# turn off shell expansion because of "?"
ps -e | grep "xyz$" | while read line
do
set $line
[ "$2" = "?" ] && kill -9 $1
done
```

If you can't do it that way, you have to get more clever, and watch for changing time:

```
set -f
mkdir /tmp/mystuff
ps -e | grep "xyz$" | while read line
do
set $line
```

```
ps -p $1 > /tmp/mystuff/first
sleep 5
#adjust sleep as necessary
ps -p $1 > /tmp/mystuff/second
diff /tmp/mystuff/first /tmp/mystuff/second || kill -9 $1
done
```

And even that may not be clever enough for your particular situation, so test and tread carefully. You may even need to do math on the time field to see what has really happened. If all else fails, you might be able to set ulimit on cpu time or some other limit that wouldn't affect a "normal" running of this process.

Bela Lubkin made an interesting post about an apparently slow CPU2 on an SMP system. Read it at /Bofcusm/1695.html.

Another thing you may see is a process that has used a lot of time but isn't gaining time right now. I've seen that many times where the process is "deliver"- MMDF's mail delivery agent on SCO systems that aren't running sendmail. What happens is that for whatever reason (a root.lock file from a crash in /usr/spool/mail or a missing "sys" home directory), there are thousands of undelivered messages in the subdirectories of /usr/spool/mmdf/lock/home

The fix for that is simple if you don't care about the messages: rm -r all those directories and recreate them empty with the same ownership and permissions

```
cd /usr/spool/mmdf/lock/home
/etc/rc2.d/P86mmdf stop
rm -r *
chown mmdf:mmdf *
chmod 777 *
cd /usr/spool/mail
rm *.lock
/etc/rc2.d/P86mmdf start
```

You'd then want to verify that mail is working normally and that whatever caused the problem isn't still happening- for example, if /usr/sys is missing this problem will come right back again very quickly.

Another possibility is a program that is rapidly spawning off other programs. You should be able to see that in "ps -e". First, are the number of processes growing?:

```
ps -e | wc -l
sleep 5
ps -e | wc -l
```

Or, are there new processes briefly showing up at the end of the listing?:

```
ps -e | tail
sleep 5
ps -e | tail
```

In either case, you need to track down the parent and kill it.

Low Memory

If sar -r shows low memory or (worse) swapping, go buy more memory. That's going to be easy to spot on SCO's sar, but Linux is a bit harder. Let's look at SCO first:

```
SCO_SV scosysv 3.2v5.0.6 PentIII    12/06/2001

11:37:06   freemem    freeswp availrmem availsmem (-r)
11:37:06        unix restarts
```

```
11:40:00    52972   786000    56222   150408
12:00:00    52903   786000    56234   150643
12:20:00    52996   786000    56240   150723
12:40:00    53018   786000    56240   150723
13:00:00    53018   786000    56240   150723
13:20:00    53018   786000    56240   150723
13:40:00    53018   786000    56240   150723
14:00:00    52885   786000    56231   150606
14:20:00    52999   786000    56240   150723
14:40:00    53016   786000    56240   150723
15:00:00    53018   786000    56240   150723
```

This machine consistently has over 200 MB of memory available- unused (freemem pages are 4K each). Obviously no problem there, and in fact, if this is always the case (which you'd know from sar historical data), you may want to use some of that memory for disk buffers- see /Unixart/memory.html.

Linux looks much different:

```
Linux 2.4.9-12 (apl)    12/06/2001

03:19:25 PM kbmemfree kbmemused  %memused kbmemshrd kbbuffers  kbcached kbswpfree kbswpused
%swpused
03:19:30 PM     4864    250080     98.09       700      3920     77136    442192     87912     16.58
03:19:35 PM     4864    250080     98.09       700      3920     77140    442192     87912     16.58
03:19:40 PM     4864    250080     98.09       700      3920     77140    442192     87912     16.58
03:19:45 PM     4864    250080     98.09       700      3920     77140    442192     87912     16.58
03:19:50 PM     4864    250080     98.09       700      3920     77140    442192     87912     16.58
Average:        4864    250080     98.09       700      3920     77139    442192     87912     16.58
```

As is immediately obvious, all memory is in use. The reason is that Linux always uses "unused" memory for the file system buffer cache. So, roughly 75 MB has been put to work for that (kbcached column).

Another way to look at Linux memory right this moment is to

```
cat /proc/meminfo
```

But what's using it? Well, ps -el will show you in the SZ column how much memory each process is using. So

```
ps -el | sort -r +9 | head
```

can tell you a lot, particularly if you see that a process is gaining memory over time.

Disk Performance

Remember, this article is not about performance tuning- it's about specific performance degradation. However, it's almost always true that the disk drives are the biggest performance bottleneck. See Raid for a more general discussion of improving disk performance.

On SCO systems, you can get a good overview of disk performance from "sar -d":

```
# sar -d 10 5

SCO_SV scosysv 3.2v5.0.6 PentIII    12/08/2001

07:57:52  device   %busy    avque    r+w/s    blks/s    avwait    avserv (-d)
07:58:02  wd-0     77.42     1.00   483.92   967.83     0.00      1.60

07:58:12  wd-0     45.45     1.00   242.36   486.11     0.01      1.88
```

```
07:58:22  wd-0     6.99     83.30     44.26    529.87    130.05      1.58

#
```

The first thing you are interested in is %busy. The "avwait" column is the average amount of time that processes wait for the disk to give them their data, so that and "avque" (how many processes are trying to use the disk) can give you a clear picture of load. The "avserve" is a measure of the disks ability to deliver that data, and isn't going to change much for the same hardware.

Note that if %busy is small and avque and avwait are high, you are probably seeing buffer cache flushes. Those can affect performance, and there are tunables that affect how often and how much is flushed, but those issues aren't the focus of this article.

Linux sar doesn't have "-d" (or at least it doesn't on Red Hat 7.2), so the next best thing is iostat:

```
iostat -d 5 2
Linux 2.4.9-12 (apl)    12/08/2001

Device:       tps   Blk_read/s   Blk_wrtn/s   Blk_read   Blk_wrtn
dev8-0       8.18        77.13        25.77     625890     209094
dev8-1       0.31         2.26         0.38      18376       3064

Device:       tps   Blk_read/s   Blk_wrtn/s   Blk_read   Blk_wrtn
dev8-0      10.80        84.80         1.60        424          8
dev8-1       0.00         0.00         0.00          0          0
```

The "tps" is the number of transfers per second the disk was asked for. The block read and write columns give you both the number of blocks and the blocks per second.

For a sudden degradation of performance, you are interested only to see if the disk is more busy than is normal. You might also be looking at the blocks per second to compare how much data is being moved around. Of course, if you decide that the disk activity is unusual, your next problem is to find out why: you still have to track down the process that is doing this.

The first place I'd look in such a situation is log files: /var/adm on SCO, or /var/log on Linux. If there's nothing there, and you see large amounts of writes, accompanied by decreasing disk space (df -v), then you are looking for a growing file and the process that is writing it. Finding that can be fairly easy:

```
cd /
du -s * > /tmp/a
sleep 30
du -s * > /tmp/bdiff /tmp/a /tmp/b
```

The directory that is growing will pop out of the diff. Change into that directory and repeat the procedure until you find the file that is growing, and finally use "fuser" to identify the process that is writing the file.

```
fuser -k /tmp/growingfile
```

Network Performance

A sudden problem with network performance is almost always going to be hardware, but there are other possibilities. For example, are your routes what they should be? If "netstat -rn" shows different routing than what you expect, do you have routed running and some router is giving you bad information? Kill routed and reset your routes.

"netstat -in" will give you an idea of collisions; a bad card somewhere in your network can cause this- if you run switches rather than hubs you won't have collisions at all, but that bad card could still be affecting performance. Know what "normal" network traffic looks like, know how long "ping" response times should be on your WAN, etc.

Mis-negotiation of network speeds and cheap nic cards are another source of network problems:

- /Bofcusm/645.html
- /SCOFAQ/scotec4.html#duplexspeed
- /SCOFAQ/scotec4.html#autonegotiate
- /Bofcusm/1235.html
- /Bofcusm/2095.html
- /Bofcusm/2096.html

An interesting mail related problem I had recently is described at /SME/largenewsletter.html

Other problems

Disk CPU and Network are the most common areas that will cause a sudden performance drop. However, there are other things that can happen. The general rule is that if you have historical data, you'll be able to spot the problem much more quickly. Some of the other things I'd look at if everything above came up blank include "sar -y" (tty activity) and "sar -q" (run queue size)- these may not be available on Linux.
