

## Re: Hard disk speed – Maybe OT

---

*Source:* <http://linux.derkeiler.com/Newsgroups/alt.os.linux.suse/2008-07/msg00623.html>

---

- *From:* Will Honea <[whonea@xxxxxxxxx](mailto:whonea@xxxxxxxxx)>
  - *Date:* Thu, 17 Jul 2008 00:42:51 -0600
- 

Bob Bob wrote:

Will/Nikos

Tnxs for your feedback. I tried to keep the question simple, but see I need to elaborate!

I have certainly allocated 2x RAM to swap but note that it is barely used. To repeat the numbers;

```
Memory: Total Used Free Shared Buffers  
Mem: 255416 138680 116736 0 90028  
Swap: 522072 8384 513688
```

Only a very small amount of swap is used and there is still 90MB of buffer space. Depending on the weather that buffer space goes up to about 120MB. This will of course all be disk cache.

I am sorry I didnt explain the system well enough. It isnt video editing and it isnt interactive. It is more a surviellance system. I capture from 6 IP cameras, jpg images at around 5-6FPS and 640x480. Each jpg compressed filesize is in the order of 50-100K. The current capture uses wget and works surprisingly well. Since I use noclobber I then have to rename the files to a timestamped name. This obviously adds time to processing that has to happen in real time, so I shell/fork out & to do this. The right way is of course to write code that captures and writes direct to a stamped filename. (I am aware that some GPL projects area available already do this) That will be the next step if I can't resolve the I/O bottleneck. The renaming process is set to inhibit the wget process if it runs overtime, which it does as the I/O load gets higher. The result is that there are gaps in the capture stream. (I use an input and output directory and swap them when renaming is finished)

After capturing I use mjpegtools (mpeg2enc etc) to create 1FPS MPEG2 videos from the individual images. (It also does motion detect) These are used to rough catch thieving events after which the 5FPS images are checked and/or a 5FPS MPEG is created for the date/time in question. We have some classic shots of people tucking shop items into their shorts,

Re: Hard disk speed – Maybe OT

pants, pockets and even socks! (Its a non profit org "thrift" store)  
Whether talking a 1FPS or 5FPS MPEG video they are between 20 and 50MB  
in size. (They are of varying time lengths)

So it isnt as badly loaded as you might think. At any one time there are  
a lot of images on the server, maybe 1–2 million for 2–3 days. They are  
broken up by camera to make find/sorts a little faster and I think NFS  
also suffers with the large number of files per directory. I cant do an  
ls for example as shell expansion runs out of space. (I use find and  
xargs) During the development (one might say playing) process I hit the  
mpeg creation limit problem first. I needed 30 hours to process the days  
data. I then clustered two more machines in and read up/implemented  
reducing the cpu grunt required for mpeg2enc. This was all originally at  
1FPS. I now find I have hit an I/O rather than cpu limit trying to  
increase the frame rate.

Apologies for the length.

I dont think RAM will help all that much. The actual I/O rate numbers  
look like;

The capture side is maybe 3Mbytes/sec but that goes to the ramdisk. The  
moving/renaming will be around the same rate but of course writes to the  
real RAID0 disk. The 1FPS mpeg creation for an hours worth of data per  
cameras takes say 15 minutes. Thats about 350MB of reads or  
0.5Mbytes/sec. There are three of these running plus a bit of other I/O  
that would on average maybe an extra 50% (I have to recreate reference  
images and that uses "convert") Say all up 3MBytes/sec. About the same  
as the capture rate. hdparm -tT for one of the RAID0 devices shows about  
20Mbytes/sec buffered disk reads. Maybe there is a lot more I/O going on  
than I thought? Have to look at the mpeg creation script some more..

Would still like to hear your ideas.

Cheers Bob

Nikos Chantziaras wrote:

With 256Mb RAM, I'm surprised that it will stagger along  
even without  
the RAM disk!

What kind of video processing  
are we talking about here exactly? Videos usually mean multi-GB file  
sizes.

I'm with Nick – I read what I expected instead of what you wrote. Now I'm a  
bit suspicious as well, even with you expanded explanation but before we  
get too far gone I'd sure like a better look at what you are doing here – I  
have a similar project in the works where your system sounds like a very

## Re: Hard disk speed – Maybe OT

good fit since there is a very cheap (about \$160 for 3 wireless cameras and a receiver that scans the 3 cameras on either a fixed schedule or on command). I think we had a brief exchange on this a few weeks back.

I started developing real time software back when we used analog computers for anything faster than about 1 hz. Given the cost of RAM circa the late 60's – early 70's we got pretty good at locating choke points and, believe it or not, RAM disk can be a killer! This is especially true as the number of files increases. What happens is that the disk cache system gets choked for lack of room. The latency of disk data (and your RAM disk is cached as well unless you've done some fiddling) is such that the cache is constantly being invalidated because it's too small! You can see this with a highly threaded compiler and build system (OS/2 was great for this) on a huge project with many, many source files producing a huge flow of object files which in turn get piped to a linker creating a large number of linked files (dll, executable, etc.). I haven't really stressed gcc, but I would expect a similar response. I've run the tests many times over the years and allowing the disk cache to use as much memory as it wants has won hands down every time. That even goes back to 4 mhz PCs with MFM/RLL drives. Once your data stream exceeds the disk cache capacity you actually increase the disk access count by nearly two to one as the over loaded cache is flushed and re-loaded for pretty much every r/m/w operation. I don't know what tools may be available for Linux but one that shows cache hit rates would probably make the problem glaringly obvious – especially for that processing loop. Try disabling the RAM disk and see if it helps.

A second trick I've used over the years that may help is to offload the disk storage to a network drive. I've had cases where that made a big difference even with 10mbs ethernet on slower machines. What you trade off here is that the disk cache is offloaded to another box so the excessive thrashing is reduced.

—

Will Honea

\*\* Posted from <http://www.teranews.com> \*\*

.