

## Re: efficacy of Linux w/o swap

**Source:** <http://linux.derkeiler.com/Newsgroups/comp.os.linux.development.system/2005-05/0256.html>

---

*phil-news-nospam\_at\_ipal.net*

**Date:** 05/17/05

Date: 17 May 2005 00:27:36 GMT

On Mon, 16 May 2005 15:24:49 GMT Robert Redelmeier <redelm@ev1.net.invalid> wrote:

| phil-news-nospam@ipal.net wrote:

> I'm wanting to put together a machine to do a single  
> dedicated purpose that will use Linux and commodity hardware.  
> The purpose would be to drive a firewire based digit video (DV)  
> encoder/decoder much like a PVR, but in this case somewhat  
> different in functionality. However, in this case, unlike a  
> PVR, performance is an absolute must. The video stream must  
> be input or output continuously without any loss. In order to  
> achieve that goal I'm expecting the process doing the actual  
> work would have to be locked into RAM to be sure it doesn't  
> ever fail to get it's needed CPU cycles to keep the stream going.

|  
| I suppose you could lock into RAM, but that eats more memory.  
| You might have to if you have some seldom-used functions that  
| still require lowlatency (cannot stand reload from disk).

RAM is relatively cheap now days. The amount I would need is for the program (which I could link statically to avoid having to lock a whole shared library) itself and for the data buffer it reads into and writes from (working as a ring buffer). Just this one program will be time critical and almost nothing else will be running on the machine.

| Personally, I set up any demanding, high performance box as  
| swapless and get ruthless on the tasklist.

We seem to be thinking along the same track.

| For something with high disk demands, I wouldn't let  
| anything pre-empt them. And the kernel swapping would.

Apparently even if there is no swap space, the kernel looks around for dirty pages to write. And in at least 2.4 it gets rather bad when memory is full and slows down. Others have said 2.6 has an inverse lookup that should help, but I don't think a single pass of all pages can account for such an extreme amount of time. Something must be causing more than that.

> When I tested these scenarios with swap disabled, things  
> actually got worse, rather than better. It seems the caching  
> still was being done, but this time the kernel choked  
> on RAM being totally full and spent large amounts of CPU  
> doing some kind of activity looking for available pages.  
> The way I observed this was through virtual console (X was  
> not running) switching. When I was in tty1 and pressed  
> Alt-F2 to go to tty2, it would pause and not switch for  
> quite some time, then eventually complete the switch.  
> Worst case event took 2.5 minutes.

|  
| Wow! I've never seen things that bad. It sounds like  
| there is \_a lot\_ of data hitting the box, and either  
| it cannot process or save it away fast enough.

This was before I started using O\_DIRECT. Maybe O\_DIRECT will avoid  
many of these problems. But I am still concerned about going swapless  
because of the fact that in the past it did make things worse, not  
better.

| You do have your partitions mounted `noatime`? This is  
| especially important with large number of files or deep  
| directories.

It happened even if the writing was to a raw partition, so it isn't  
just "noatime", although using that option can certainly decrease the  
amount of disk operations needed. I use it everywhere except on  
/var/spool/mail for reasons of getting correct mailbox change detection.

--

-----  
| Phil Howard KA9WGN | <http://linuxhomepage.com/> | <http://ham.org/> |  
| (first name) at ipal.net | <http://phil.ipal.org/> | <http://ka9wgn.ham.org/> |  
-----