

Re: serious networking (em) performance (ggate and NFS) problem

Source: <http://unix.derkeiler.com/Mailing-Lists/FreeBSD/stable/2004-11/0780.html>

From: Andre Oppermann (*andre_at_freebsd.org*)

Date: 11/29/04

Date: Mon, 29 Nov 2004 16:15:48 +0100

To: "David G. Lawrence" <dg@dglawrence.com>

"David G. Lawrence" wrote:

>
> > > tests. With the re driver, no change except placing a 100BT setup with
> > > no packet loss to a gigE setup (both linksys switches) will cause
> > > serious packet loss at 20Mbps data rates. I have discovered the only
> > > way to get good performance with no packet loss was to
> > >
> > > 1) Remove interrupt moderation
> > > 2) defrag each mbuf that comes in to the driver.
> > >
> > > Sounds like you're bumping into a queue limit that is made worse by
> > > interrupting less frequently, resulting in bursts of packets that are
> > > relatively large, rather than a trickle of packets at a higher rate.
> > > Perhaps a limit on the number of outstanding descriptors in the driver or
> > > hardware and/or a limit in the netisr/ifqueue queue depth. You might try
> > > changing the default IFQ_MAXLEN from 50 to 128 to increase the size of the
> > > ifnet and netisr queues. You could also try setting net.isr.enable=1 to
> > > enable direct dispatch, which in the in-bound direction would reduce the
> > > number of context switches and queueing. It sounds like the device driver
> > > has a limit of 256 receive and transmit descriptors, which one supposes is
> > > probably derived from the hardware limit, but I have no documentation on
> > > hand so can't confirm that.
> > >
> > > It would be interesting on the send and receive sides to inspect the
> > > counters for drops at various points in the network stack; i.e., are we
> > > dropping packets at the ifq handoff because we're overfilling the
> > > descriptors in the driver, are packets dropped on the inbound path going
> > > into the netisr due to over-filling before the netisr is scheduled, etc.
> > > And, it's probably interesting to look at stats on filling the socket
> > > buffers for the same reason: if bursts of packets come up the stack, the
> > > socket buffers could well be being over-filled before the user thread can
> > > run.
> > >
> > > I think it's the tcp_output() path that overflows the transmit side of
> > > the card. I take that from the better numbers when he defrags the packets.

freebsd-stable: Re: serious networking (em) performance (ggate and NFS) problem

- > > *Once I catch up with my mails I start to put up the code I wrote over the*
- > > *last two weeks. :-) You can call me Mr. TCP now. ;-)*
- >
- > *He was doing his test with NFS over TCP, right? ...That would be a single*
- > *connection, so how is it possible to 'overflow the transmit side of the*
- > *card'? The TCP window size will prevent more than 64KB to be outstanding.*
- > *Assuming standard size ethernet frames, that would be a maximum of 45 packets*
- > *in-flight at any time (65536/1460=45), well below the 256 available transmit*
- > *descriptors.*
- > *It is also worth pointing out that 45 full-size packets is 540us at*
- > *gig-e speeds. Even when you add up typical switch latencies and interrupt*
- > *overhead and coalescing on both sides, it's hard to imagine that the window*
- > *size (bandwidth * delay) would be a significant limiting factor across a*
- > *gig-e LAN.*

For some reason he is getting long mbuf chains and that is why a call to m_defrag helps. With long mbuf chains you can easily overflow the transmit descriptors.

- > *I too am seeing low NFS performance (both TCP and UDP) with non-SMP*
- > *5.3, but on the same systems I can measure raw TCP performance (using*
- > *ttcp) of >850Mbps. It looks to me like there is something wrong with*
- > *NFS, perhaps caused by delays with scheduling nfsd?*

--
Andre

freebsd-stable@freebsd.org mailing list
<http://lists.freebsd.org/mailman/listinfo/freebsd-stable>
To unsubscribe, send any mail to "freebsd-stable-unsubscribe@freebsd.org"