

Re: Removing T/TCP and replacing it with something simpler

Source: <http://unix.derkeiler.com/Mailing-Lists/FreeBSD/arch/2004-10/0188.html>

From: Brian Fundakowski Feldman (green_at_freebsd.org)

Date: 10/22/04

Date: Fri, 22 Oct 2004 11:45:17 -0400
To: Andre Oppermann <andre@freebsd.org>

On Fri, Oct 22, 2004 at 05:14:07PM +0200, Andre Oppermann wrote:

> *Sean Chittenden wrote:*

>>

>>>> *However something like T/TCP is certainly useful and I know of one*

>>>> *special*

>>>> *purpose application using it (Web Proxy Server/Client for high-delay*

>>>> *Satellite*

>>>> *connections).*

>>>>

>>>> *Actually, there are two/three programs that I know of that use it.*

>>>> *memcached(1), which found a fantastic decrease in its benchmarks.*

>>>> *Here's an excerpt from the following link:*

>>>>

>>>> <http://lists.danga.com/pipermail/memcached/2003-August/000111.html>

>>>>

>>>> *I think you got something wrong here. T/TCP is never ever mentioned*

>>>> *in this. Memcached is not using T/TCP as far as I can see.*

>>>>

>>>> *It's not, but I thought setsockopt(2) w/ TCP_NOPUSH enabled the use of*

>>>> *T/TCP in that there was no 3WS performed on a TCP_NOPUSH'ed*

>>>> *connection.*

>>>>

>>>> *No, it is not. T/TCP will only be used if you use sendto(), have T/TCP*

>>>> *globally enabled on the machine and the server supports it too.*

>>>>

>>>> *TCP_NOPUSH was introduced together with or some time after T/TCP to*

>>>> *change the behaviour how tcp_output() pushes non-full packets on the*

>>>> *wire. It pretty closely related to the same purpose as TCP_CORK.*

>>>>

>>>> *and an internal reverse proxy server/modified apache that I've hacked*

>>>> *together (reduces latency in a tiered request hierarchy a great deal,*

>>>> *on order of the benchmarks from above).*

>>>>

>>>> *What syscall do you use to get to the other side in your reverse proxy?*

>>>>

freebsd-arch: Re: Removing T/TCP and replacing it with something simpler

> > *On the client, sendto()/read(). On the server, setsockopt() + write().*
>
> *Ok, then you are indeed using T/TCP (provided you have enabled it on*
> *both machines). The setsockopt() optimizes packet sending on the server*
> *but otherwise doesn't have anything to do with T/TCP.*
>
> > > *I'm not sure if I can follow you here. TCP_CORK deals with the*
> > > *different*
> > > *behaviour of connections with Nagle vs. TCP_NODELAY. TCP_CORK allows*
> > > *to*
> > > *avoid the delays of Nagle by corking (sort of blocking) the sending of*
> > > *packets until you are done with write()ing to the socket. Then the*
> > > *connection is uncorked and all data will be sent in one go even if it*
> > > *doesn't fill an entire packet. Sort of an fsync() for sockets. There*
> > > *are no security implications with TCP_CORK as far as I am aware.*
> >
> > *Isn't that what NOPUSH does? Or is it that CORK uses a fully*
> > *established TCP connection, but blocks sending data until the*
> > *connection has been uncorked/flushed? I thought that TCP_CORK had the*
> > *same security implications that NOPUSH does (ie, the lack of a hand*
> > *shake).*
>
> *None of it. Neither NOPUSH nor CORK have any security implications.*
> *Those are only with the specification of T/TCP. Blocking the data*
> *is independend of 3WSH. Normally you have Nagle enabled (default)*
> *and when you don't fill an entire packet worth of data it will wait*
> *up to 200ms to send the packet in anticipation of more data from the*
> *socket. This screws the responsiveness of your connection. The first*
> *solution is to turn off Nagle (with TCP_NODELAY) but now you get a*
> *packet for every single write() you do. Fine for telnet and ssh but*
> *not the right thing for a database server. There you don't want the*
> *delay but at the same time you want several successive write()s that*
> *will go in one packet on the wire. Here NOPUSH and CORK come into*
> *play.*

Why is just tuning the delay a bad solution?

--
Brian Fundakowski Feldman
<> green@FreeBSD.org
Opinions expressed are my own.

```
\'[ FreeBSD ]'''''''''''\n \ The Power to Serve! \n \.....\n
```

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