

Re: sendfile(2) SF_NOPUSH flag proposal

Source: <http://unix.derkeiler.com/Mailing-Lists/FreeBSD/arch/2003-05/0079.html>

From: Peter Jeremy (peterjeremy_at_optushome.com.au)

Date: 05/26/03

Date: Tue, 27 May 2003 06:17:41 +1000

To: Igor Sysoev <is@rambler-co.ru>

On Mon, May 26, 2003 at 09:41:50PM +0400, Igor Sysoev wrote:

> *sendfile(2)* now has two drawbacks:

[IP frames are not always full]

...

> When I turn *TCP_NOPUSH* on just before *sendfile()* then it sends the header
> and the first part of the file in one 1460 bytes packet.

> Besides it sends file pages in the full ethernet 1460 bytes packets.

> When *sendfile()* completed or returned *EAGAIN* (I use non-blocking sockets)

> I turn *TCP_NOPUSH* off and the remaining file part is flushed to client.

> Without turning off the remaining file part is delayed for 5 seconds.

...

> So here is a proposal. We can introduce a *sendfile(2)* flag, i.e. *SF_NOPUSH*

> that will turn *TF_NOPUSH* on before the sending and turn it off just

> before return. It allows to save two syscalls on each *sendfile()* call

> and it's especially useful with non-blocking sockets – they can cause many

> *sendfile()* calls.

I'm less certain of the benefits of this – particularly in the non-blocking case. As I understand your proposal, your patch would turn off *TF_NOPUSH* just before returning *EAGAIN*. At this point, the TCP send buffer is full so packets should start being sent immediately.

The last data in the send buffer may not comprise a complete frame so it should not be sent, but left queued to be merged with the next *sendfile(2)*. Once *SO_SNDLOWAT* bytes are available in the send buffer, the socket will become writable, allowing a further *sendfile(2)* call.

As long as *SO_SNDLOWAT* is at least one frame smaller than *SO_SNDBUF*, there should not be any send delay caused by *TF_NOPUSH* being set.

I believe *TF_NOPUSH* should be set at the beginning of a transaction (or when the socket is opened) and cleared at the end of a transaction (or implicitly by *close()*ing the socket).

Peter

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"