

Re: [6.x] problem with AIO, non-blocking sockets on freebsd and IE7 on windows.

Re: [6.x] problem with AIO, non-blocking sockets on freebsd and IE7 on windows.

Source: <http://unix.derkeiler.com/Mailing-Lists/FreeBSD/net/2007-06/msg00198.html>

- *From:* Bill Moran <wmoran@xxxxxxxxxxxxxxxxxxxxxxxxxxxxx>
 - *Date:* Mon, 25 Jun 2007 13:31:36 -0400
-

In response to Julian Elischer <julian@xxxxxxxxxxxxx>:

Bruce Evans wrote:

On Fri, 22 Jun 2007, Julian Elischer wrote:

If one has an event-driven process that accepts tcp connections, one needs to set eh non-blocking socket option and use kqueue or similar to schedule work.

This is ok for data transfers, however when it comes to the close() call there is a problem. The problem is in the following code in so_close()

```
if (so->so_options & SO_LINGER) {
if ((so->so_state & SS_ISDISCONNECTING) &&
(so->so_state & SS_NBIO))
goto drop;
...
drop:
[ continues on to destroy socket ]
```

because SS_NBIO is set, the socket acts as if SO_LINGER was set, with a timeout of 0. the result of this, is the following behaviour:

[packets in flight get lost]

Re: [6.x] problem with AIO, non-blocking sockets on freeBSD and IE7 on windows.

This seems to be the correct behaviour. The application doesn't care about its data and/or wants to close the descriptor without blocking, so it doesn't turn off the blocking flag and/or wait for i/o to complete (so that it can see if the i/o actually worked) before calling close().

It's not the correct behaviour if the only packet coming back is an Ack of the FIN (and a FIN) because in the real world, making IE7 throw an error screen is not an acceptable option. This is the sort of thing that gets FreeBSD thrown out on favour of "anything else". Believe me, our customers are "NOT HAPPY" about this. Instead of getting an "authorization required" page along with the opportunity to log in, they get an error, and no opportunity to log in, which makes the system unusable. Yes, Blame Microsoft, but we are breaking the TCP spec, not them. We need to fix this some how.

Correct me if I'm wrong, but I took Bruce's explanation to mean that the problem is not in FreeBSD, it's the fault of the application not using non-blocking IO in the manner it was intended.

If you want properly closed connections, you turn of non-blocking before you close the connection. If you want fast close that's not contingent on anything, you close non-blocking and accept that some data may be lost and some errors may result.

If you tell the OS to drop packets and it drops them, the OS isn't in error for following orders.

Or did I misunderstand?

--

Bill Moran
Collaborative Fusion Inc.
<http://people.collaborativefusion.com/~wmoran/>

wmoran@xxxxxxxxxxxxxxxxxxxxxxxxxxxxx
Phone: 412-422-3463x4023

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