

Re: kqueue disable on delivery...

Source: <http://unix.derkeiler.com/Mailing-Lists/FreeBSD/arch/2006-09/msg00067.html>

- *From:* John-Mark Gurney <gurney_j@xxxxxxxxxxxxxxxxxxxxxx>
 - *Date:* Fri, 22 Sep 2006 09:58:49 -0700
-

Igor Sysoev wrote this message on Fri, Sep 22, 2006 at 17:25 +0400:

On Sun, 17 Sep 2006, John-Mark Gurney wrote:

I have implemented a couple additional features to kqueue. These allow kqueue to be a multithreaded event delivery system that can guarantee that the event will only be active in one thread at any time.

The first is EV_DOD, aka disable on delivery. When the event will be delivered to userland, the knote is marked disabled so we don't have to go through the expense of reallocating the knote each time. (Reallocation of the knote is also lock intensive, and disabling is cheap.)

In my opinion, it's too implementation specific flag.

How else are you doing to solve having multiple threads servicing the same queue at the same time? Also, Apple is planing on having a similar flag to EV_DOD, but I don't know what they are naming it.. I've tried for a while to find out, but haven't been able to...

Even though this means that the event will only ever be active in a thread at a time, (when you're done handling the event, you reenable it), removing the event from the queue outside the event handler (say a timeout handler for the connection) poses to be a problem. If you simply close the socket, the event disappears, but then there is a race between another event being created with the same socket, and notification of the handler that you want the event to stop.

In order to handle that situation, I have come up w/ EV_FORCEOS, aka FORCE ONE_SHOT. EV_ONESHOT events have the advantage that once queued, they don't care if they have been activated or not, they will be returned the next round. This means that the timeout handler can safely set EV_FORCEOS on the handler, and either if it's _DISABLED (handler

Re: kqueue disable on delivery...

running
and will reenable it), or it's `_ENABLED`, it will get dispatched, allowing
the handler to detect the `EV_FORCEOS` flag and teardown the connection.

I think it should be `EVFILT_USER` event, allowing to
`EV_SET(&kev, fd, EVFILT_USER, 0, 0, 0, udata);`
and the event should automatically sets the `EV_ONESHOT` flag internally.

I'll agree `EV_FORCEOS` is open for discussion, but you did see how much
code it adds right? I was surprised at how small the patch was for the
additional functionality..

What happens if you are in the process of tearing down `udata` when
this happens, but you haven't gotten far enough to drop it? Then
you'd have to deal w/ possible lock inversions between the timeout
list and your object lock, deal w/ flags on the object and ref counts..

With `_DOD` and `_FORCEOS`, you are able to continue to not require special
state flags, locks nor reference counting on your objects serviced by
kqueue...

I wrote this code in anticipation of supporting `sun4v` boxes where it'd
be useful to have 32 threads (or more) servicing a single kqueue...

John-Mark Gurney Voice: +1 415 225 5579

"All that I will do, has been done, All that I have, has not."

freebsd-arch@xxxxxxxxxxxxx mailing list

<http://lists.freebsd.org/mailman/listinfo/freebsd-arch>

To unsubscribe, send any mail to "freebsd-arch-unsubscribe@xxxxxxxxxxxxx"