

Re: Call for performance evaluation: net.isr.direct

Source: <http://unix.derkeiler.com/Mailing-Lists/FreeBSD/performance/2005-10/0019.html>

From: Robert Watson (rwatson_at_FreeBSD.org)

Date: 10/12/05

Date: Wed, 12 Oct 2005 13:33:06 +0100 (BST)

To: gnn@freebsd.org

On Wed, 12 Oct 2005 gnn@freebsd.org wrote:

> At Tue, 11 Oct 2005 15:01:11 +0100 (BST),
> *rwatson* wrote:
>> *If I don't hear anything back in the near future, I will commit a
>> change to 7.x to make direct dispatch the default, in order to let a
>> broader community do the testing. :-)* *If you are setup to easily
>> test stability and performance relating to direct dispatch, I would
>> appreciate any help.*
>
> *One thing I would caution, though I have no proof nor have I made any
> tests (yes, I know, bad gnn), is that I would expect this change to
> degrade non-network performance when the network is under load. This
> kind of change is most likely to help those with purely network loads,
> i.e. routers, bridges, etc and to hurt anyone else. Are you absolutely
> sure we should make this the default?*

In theory, as I mentioned in my earlier e-mail, this does result in more network processing occurring at a hardware ithread priority. However, the software ithread (swi) priority is already quite high. Looking closely at that is probably called for — specifically, how will this impact scheduling for other hardware (rather than software) ithreads?

The most interesting effect I've seen on non-network applications is that, because the network stack now uses significantly less CPU when under high load, more CPU is available for other activities. With the performance of network hardware available on server now often exceeding the CPU capacity of those servers (as compared to a few years ago when 100mbps cards could be trivially saturated by server hardware), the cost of processing packets is now back up again, so this can occur with relative ease. Another interesting point is that remote traffic can now no longer result in a denial of service of local traffic by virtue of overflowing the netisr queue. Previously, a single queue was shared by all network interfaces going to the netisr, and in the direct dispatch model, the queueing now happens almost entirely in the device driver and skips entering a queue to get to the stack. This has some other interesting effects, not least that

freebsd-performance: Re: Call for performance evaluation: net.isr.direct

older cards with less buffering now see significantly less queue space, but I'm not sure if that's significant.

In general, I agree with your point though: we need to evaluate the effect of this change on a broad array of real-world workloads. Hence my e-mail, which so far has seen two responses — a private one from Mike Tancsa offering to run testing, and your public one. So anyone willing to help evaluate the performance of this change would be most welcome to.

Robert N M Watson

freebsd-performance@freebsd.org mailing list

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

To unsubscribe, send any mail to "freebsd-performance-unsubscribe@freebsd.org"