

Re: MPSAFE CAM?

Source: <http://unix.derkeiler.com/Mailing-Lists/FreeBSD/arch/2005-05/0116.html>

From: Scott Long (scottl_at_samsco.org)

Date: 05/28/05

Date: Sat, 28 May 2005 15:31:02 -0600

To: Daniel Eriksson <daniel_k_eriksson@telia.com>

Daniel Eriksson wrote:

> *I'm sorry to hijack your thread Robert, but I just wanted to hear if there*
> *is any progress in making CAM (and the individual card drivers) MPSAFE?*
>
> *FreeBSD is mainly used for servers, where SCSI is pretty common. To me it*
> *seems like locking down CAM would be beneficial to a large number of FreeBSD*
> *users.*
>
> *I don't have the knowledge or time to do any real work on this myself, but I*
> *should be able to help test patches.*
>
> */Daniel Eriksson*
>
>

CAM was originally designed to make *_very_* efficient use of SPL synchronization. Simply dropping locks in place of spl's has been tried, and the result is horrible performance. It's going to take a significant rewrite of much of the XPT code to make it efficiently locked. I started on this about a year ago but got sidetracked. It's still very high on my priority list, and I hope that I can get to it this fall. In the mean time, I put in some minimal locking so that `xpt_done()` can be called from drivers without Giant being held. This means that drivers can lock their interrupt service thread and declare themselves MPSAFE. The top half of the driver will still run with Giant held when it's called from CAM, but the bottom half can run Giant-free. I modified the `esp(4)` driver a few months ago to demonstrate this, and the results seem to be ok. `esp` definitely benefits from it since its scsi state machine is in the driver, so 4-5 interrupts get generated for each I/O. For more modern SCSI chips like `ahc/ahd` and `mpt`, the benefits might not be as much.

If anyone wants to help tackle this, I'd be glad to help them get started. BSD/OS 5.(mumble) did a minimal locking job on CAM that probably works OK for systems with a single SCSI port, but likely gets very inefficient once you go beyond that. They also rewrote the

freebsd-arch: Re: MPSAFE CAM?

SCSI probe state machine, which in our CAM is a source for quite a bit of lock recursion. The work I did last year focused on addressing this.

Scott

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"