Subject: Re: [PATCH] sas_scsi_host: Convert to use the kthread API
Posted by akpm on Fri, 20 Apr 2007 00:37:53 GMT
View Forum Message <> Reply to Message

On Thu, 19 Apr 2007 01:58:38 -0600
"Eric W. Biederman" <ebiederm@xmission.com> wrote:

> From: Eric W. Biederman <ebiederm@xmission.com>
>
> This patch modifies the sas scsi host thread startup
> to use kthread_run not kernel_thread and deamonize.
> kthread_run is slightly simpler and more maintainable.
>

Again, I'll rename this to "partially convert...". This driver should be using kthread_should_stop() and kthread_stop() rather than the apparently-unnecessary ->queue_thread_kill thing.

This driver was merged two and a half years after the kthread API was available. Our coding-vs-reviewing effort is out of balance.

> ---
> drivers/scsi/libsas/sas_scsi_host.c | 11 ++++++-----
> 1 files changed, 6 insertions(+), 5 deletions(-)
>
> diff --git a/drivers/scsi/libsas/sas_scsi_host.c b/drivers/scsi/libsas/sas_scsi_host.c
> index 46ba3a7..7a38ac5 100644
> --- a/drivers/scsi/libsas/sas_scsi_host.c
> +++ b/drivers/scsi/libsas/sas_scsi_host.c
> @@ -40,6 +40,7 @@
> #include <linux/blkdev.h>
> #include <linux/scatterlist.h>
> #include <linux/freezer.h>
> +#include <linux/kthread.h>
>
> /* ---------- SCSI Host glue ---------- */
>
> @@ -870,7 +871,6 @@ static int sas_queue_thread(void *_sas_ha)
> struct sas_ha_struct *sas_ha = _sas_ha;
> struct scsi_core *core = &sas_ha->core;
>
> - daemonize("sas_queue_%d", core->shost->host_no);
> current->flags |= PF_NOFREEZE;
>
> complete(&queue_th_comp);
> @@ -891,19 +891,20 @@ static int sas_queue_thread(void *_sas_ha)
>

```
>  int sas_init_queue(struct sas_ha_struct *sas_ha)
>  {
> - int res;
>   struct scsi_core *core = &sas_ha->core;
> + struct task_struct *task;
>
>   spin_lock_init(&core->task_queue_lock);
>   core->task_queue_size = 0;
>   INIT_LIST_HEAD(&core->task_queue);
>   init_MUTEX_LOCKED(&core->queue_thread_sema);
>
> - res = kernel_thread(sas_queue_thread, sas_ha, 0);
> - if (res >= 0)
> + task = kthread_run(sas_queue_thread, sas_ha,
> +      "sas_queue_%d", core->shost->host_no);
> + if (!IS_ERR(task))
>    wait_for_completion(&queue_th_comp);
>
> - return res < 0 ? res : 0;
> + return IS_ERR(task) ? PTR_ERR(task) : 0;
```

Does that wait_for_completion(&queue_th_comp) actually do anything useful?

If so, what is serialising access to the single queue_th_comp?

_____
Containers mailing list
Containers@lists.linux-foundation.org
https://lists.linux-foundation.org/mailman/listinfo/containers