
Subject: Re: [PATCH] sas_scsi_host: Convert to use the kthread API

Posted by [Christoph Hellwig](#) on Sun, 22 Apr 2007 19:38:43 GMT

[View Forum Message](#) <> [Reply to Message](#)

On Thu, Apr 19, 2007 at 05:37:53PM -0700, Andrew Morton wrote:

> 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 daemonize.

> > 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.

Here's a full conversion.

Signed-off-by: Christoph Hellwig <hch@lst.de>

Index: linux-2.6/drivers/scsi/libsas/sas_scsi_host.c

```
=====
--- linux-2.6.orig/drivers/scsi/libsas/sas_scsi_host.c 2007-04-22 20:30:39.000000000 +0200
+++ linux-2.6/drivers/scsi/libsas/sas_scsi_host.c 2007-04-22 20:36:51.000000000 +0200
@@ -23,6 +23,8 @@
 *
 */
```

```
+#include <linux/kthread.h>
```

```
+
```

```
#include "sas_internal.h"
```

```
#include <scsi/scsi_host.h>
```

```
@@ -184,7 +186,7 @@ static int sas_queue_up(struct sas_task
```

```
list_add_tail(&task->list, &core->task_queue);
```

```
core->task_queue_size += 1;
```

```
spin_unlock_irqrestore(&core->task_queue_lock, flags);
```

```
- up(&core->queue_thread_sema);
```

```
+ wake_up_process(core->queue_thread);
```

```
return 0;
```

```

}
@@ -819,7 +821,7 @@ static void sas_queue(struct sas_ha_stru
    struct sas_internal *i = to_sas_internal(core->shost->transportt);

    spin_lock_irqsave(&core->task_queue_lock, flags);
- while (!core->queue_thread_kill &&
+ while (!kthread_should_stop() &&
        !list_empty(&core->task_queue)) {

    can_queue = sas_ha->lld_queue_size - core->task_queue_size;
@@ -858,8 +860,6 @@ static void sas_queue(struct sas_ha_stru
    spin_unlock_irqrestore(&core->task_queue_lock, flags);
}

-static DECLARE_COMPLETION(queue_th_comp);
-
/**
 * sas_queue_thread -- The Task Collector thread
 * @_sas_ha: pointer to struct sas_ha
@@ -867,40 +867,33 @@ static DECLARE_COMPLETION(queue_th_comp)
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);
-
    while (1) {
- down_interruptible(&core->queue_thread_sema);
+ set_current_state(TASK_INTERRUPTIBLE);
+ schedule();
        sas_queue(sas_ha);
- if (core->queue_thread_kill)
+ if (kthread_should_stop())
            break;
    }

- complete(&queue_th_comp);
-
    return 0;
}

int sas_init_queue(struct sas_ha_struct *sas_ha)
{
- int res;

```

```

struct scsi_core *core = &sas_ha->core;

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)
- wait_for_completion(&queue_th_comp);
-
- return res < 0 ? res : 0;
+ core->queue_thread = kthread_run(sas_queue_thread, sas_ha,
+ "sas_queue_%d", core->shost->host_no);
+ if (IS_ERR(core->queue_thread))
+ return PTR_ERR(core->queue_thread);
+ return 0;
}

void sas_shutdown_queue(struct sas_ha_struct *sas_ha)
@@ -909,10 +902,7 @@ void sas_shutdown_queue(struct sas_ha_struct
    struct scsi_core *core = &sas_ha->core;
    struct sas_task *task, *n;

- init_completion(&queue_th_comp);
- core->queue_thread_kill = 1;
- up(&core->queue_thread_sema);
- wait_for_completion(&queue_th_comp);
+ kthread_stop(core->queue_thread);

    if (!list_empty(&core->task_queue))
        SAS_DPRINTK("HA: %llx: scsi core task queue is NOT empty!?\n",
Index: linux-2.6/include/scsi/libsas.h
=====
--- linux-2.6.orig/include/scsi/libsas.h 2007-04-22 20:32:41.000000000 +0200
+++ linux-2.6/include/scsi/libsas.h 2007-04-22 20:32:59.000000000 +0200
@@ -314,8 +314,7 @@ struct scsi_core {
    struct list_head task_queue;
    int task_queue_size;

- struct semaphore queue_thread_sema;
- int queue_thread_kill;
+ struct task_struct *queue_thread;
};

struct sas_ha_event {

```

Containers mailing list

