Subject: i2o: CONFIG_DEBUG_SG compilation fixed
Posted by vaverin on Mon, 29 Oct 2007 06:32:13 GMT
View Forum Message <> Reply to Message

i2o crashed when CONFIG_DEBUG_SG is enabled because i2o_block_request structure
includes array of scatterlists that should be initialised

Signed-off-by: Vasily Averin <vvs@sw.ru>

```
--- a/drivers/message/i2o/i2o_block.c
+++ b/drivers/message/i2o/i2o_block.c
@@ -1137,6 +1137,18 @@ static struct i2o_driver i2o_block_driver = {
  *
  * Returns 0 on success or negative error code on failure.
  */
+
+#ifdef CONFIG_DEBUG_SG
+static void i2o_block_req_ctor(struct kmem_cache *cachep, void *objp) {
+ sg_init_table(((struct i2o_block_request *)(objp))->sg_table,
+       I2O_MAX_PHYS_SEGMENTS);
+}
+
+#define I2O_BLK_CTOR &i2o_block_req_ctor
+#else
+#define I2O_BLK_CTOR NULL
+#endif
+
 static int __init i2o_block_init(void)
 {
  int rc;
@@ -1147,7 +1159,7 @@ static int __init i2o_block_init(void)
  /* Allocate request mempool and slab */
  size = sizeof(struct i2o_block_request);
  i2o_blk_req_pool.slab = kmem_cache_create("i2o_block_req", size, 0,
-       SLAB_HWCACHE_ALIGN, NULL);
+       SLAB_HWCACHE_ALIGN, I2O_BLK_CTOR);
  if (!i2o_blk_req_pool.slab) {
   osm_err("can't init request slab\n");
   rc = -ENOMEM;
```

Subject: Re: i2o: CONFIG_DEBUG_SG compilation fixed
Posted by Jens Axboe on Mon, 29 Oct 2007 08:58:44 GMT
View Forum Message <> Reply to Message

On Mon, Oct 29 2007, Vasily Averin wrote:
> i2o crashed when CONFIG_DEBUG_SG is enabled because i2o_block_request structure
> includes array of scatterlists that should be initialised
>

> Signed-off-by: Vasily Averin <vvs@sw.ru>
> --- a/drivers/message/i2o/i2o_block.c
> +++ b/drivers/message/i2o/i2o_block.c
> @@ -1137,6 +1137,18 @@ static struct i2o_driver i2o_block_driver = {
> *
> * Returns 0 on success or negative error code on failure.
> */
> +
> +#ifdef CONFIG_DEBUG_SG
> +static void i2o_block_req_ctor(struct kmem_cache *cachep, void *objp) {
> + sg_init_table(((struct i2o_block_request *)(objp))->sg_table,
> + I2O_MAX_PHYS_SEGMENTS);
> +}
> +
> +#define I2O_BLK_CTOR &i2o_block_req_ctor
> +#else
> +#define I2O_BLK_CTOR NULL
> +#endif
> +
> static int __init i2o_block_init(void)
> {
> int rc;
> @@ -1147,7 +1159,7 @@ static int __init i2o_block_init(void)
> /* Allocate request mempool and slab */
> size = sizeof(struct i2o_block_request);
> i2o_blk_req_pool.slab = kmem_cache_create("i2o_block_req", size, 0,
> - SLAB_HWCACHE_ALIGN, NULL);
> + SLAB_HWCACHE_ALIGN, I2O_BLK_CTOR);
> if (!i2o_blk_req_pool.slab) {
> osm_err("can't init request slab\n");
> rc = -ENOMEM;

This should already work, since Oct 24. Your i2o_block_request_alloc()
should look like this:

```
static inline struct i2o_block_request *i2o_block_request_alloc(void)
{
	struct i2o_block_request *ireq;

	ireq = mempool_alloc(i2o_blk_req_pool.pool, GFP_ATOMIC);
	if (!ireq)
		return ERR_PTR(-ENOMEM);

	INIT_LIST_HEAD(&ireq->queue);
	sg_init_table(ireq->sg_table, I2O_MAX_PHYS_SEGMENTS);

	return ireq;
}
```

Note that I also don't like your solution, no need to change this to be
a constructor setup and you definitely should not guard sg_init_table()
with CONFIG_DEBUG_SG. It needs to be done always.

--
Jens Axboe
