Subject: dm: bounce_pfn limit added
Posted by vaverin on Mon, 29 Oct 2007 06:31:39 GMT
View Forum Message <> Reply to Message

Device mapper uses its own bounce_pfn that may differ from one on underlying
device. In that way dm can build incorrect requests that contain sg elements
greater than underlying device is able to handle.

This is the cause of slab corruption in i2o layer, occurred on i386 arch when
very long direct IO requests are addressed to dm-over-i2o device.

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

```
--- a/drivers/md/dm-table.c
+++ b/drivers/md/dm-table.c
@@ -102,6 +102,8 @@ static void combine_restrictions_low(struct io_restrictions
  lhs->seg_boundary_mask =
   min_not_zero(lhs->seg_boundary_mask, rhs->seg_boundary_mask);

+ lhs->bounce_pfn = min_not_zero(lhs->bounce_pfn, rhs->bounce_pfn);
+
  lhs->no_cluster |= rhs->no_cluster;
 }

@@ -566,6 +568,8 @@ void dm_set_device_limits(struct dm_target *ti, struct
   min_not_zero(rs->seg_boundary_mask,
        q->seg_boundary_mask);

+ rs->bounce_pfn = min_not_zero(rs->bounce_pfn, q->bounce_pfn);
+
  rs->no_cluster |= !test_bit(QUEUE_FLAG_CLUSTER, &q->queue_flags);
 }
 EXPORT_SYMBOL_GPL(dm_set_device_limits);
@@ -707,6 +711,8 @@ static void check_for_valid_limits(struct io_restrictions
   rs->max_segment_size = MAX_SEGMENT_SIZE;
  if (!rs->seg_boundary_mask)
   rs->seg_boundary_mask = -1;
+ if (!rs->bounce_pfn)
+  rs->bounce_pfn = -1;
 }

 int dm_table_add_target(struct dm_table *t, const char *type,
@@ -891,6 +897,7 @@ void dm_table_set_restrictions(struct dm_table *t, struct
  q->hardsect_size = t->limits.hardsect_size;
  q->max_segment_size = t->limits.max_segment_size;
  q->seg_boundary_mask = t->limits.seg_boundary_mask;
+ q->bounce_pfn = t->limits.bounce_pfn;
  if (t->limits.no_cluster)
```

```
   q->queue_flags &= ~(1 << QUEUE_FLAG_CLUSTER);
  else
--- a/include/linux/device-mapper.h
+++ b/include/linux/device-mapper.h
@@ -111,6 +111,7 @@ struct target_type {

 struct io_restrictions {
  unsigned long  seg_boundary_mask;
+ unsigned long  bounce_pfn;
  unsigned int  max_sectors;
  unsigned int  max_segment_size;
  unsigned short  max_phys_segments;
```