Subject: [PATCH]: OOM can panic due to processes stuck in __alloc_pages()
Posted by Kirill Korotaev on Mon, 13 Nov 2006 19:05:42 GMT
View Forum Message <> Reply to Message

OOM can panic due to the processes stuck in __alloc_pages()
doing infinite rebalance loop while no memory can be reclaimed.
OOM killer tries to kill some processes, but unfortunetaly,
rebalance label was moved by someone below the TIF_MEMDIE check,
so buddy allocator doesn't see that process is OOM-killed
and it can simply fail the allocation :/

Observed in reality on RHEL4(2.6.9)+OpenVZ kernel when a user doing
some memory allocation tricks triggered OOM panic.

Signed-Off-By: Denis Lunev <den@sw.ru>
Signed-Off-By: Kirill Korotaev <dev@openvz.org>

```
--- ./mm/page_alloc.c.oomx 2006-11-08 17:44:16.000000000 +0300
+++ ./mm/page_alloc.c 2006-11-13 21:57:33.000000000 +0300
@@ -1251,6 +1251,7 @@ restart:

  /* This allocation should allow future memory freeing. */

+rebalance:
  if (((p->flags & PF_MEMALLOC) || unlikely(test_thread_flag(TIF_MEMDIE)))
    && !in_interrupt()) {
   if (!(gfp_mask & __GFP_NOMEMALLOC)) {
@@ -1272,7 +1273,6 @@ nofail_alloc:
  if (!wait)
   goto nopage;

-rebalance:
  cond_resched();

  /* We now go into synchronous reclaim */
```

Subject: Re: [PATCH]: OOM can panic due to processes stuck in __alloc_pages()
Posted by Andrew Morton on Mon, 13 Nov 2006 22:56:38 GMT
View Forum Message <> Reply to Message

On Mon, 13 Nov 2006 22:13:47 +0300
Kirill Korotaev <dev@openvz.org> wrote:

> OOM can panic due to the processes stuck in __alloc_pages()
> doing infinite rebalance loop while no memory can be reclaimed.
> OOM killer tries to kill some processes, but unfortunetaly,
> rebalance label was moved by someone below the TIF_MEMDIE check,

```
> so buddy allocator doesn't see that process is OOM-killed
> and it can simply fail the allocation :/
>
> Observed in reality on RHEL4(2.6.9)+OpenVZ kernel when a user doing
> some memory allocation tricks triggered OOM panic.
>
> Signed-Off-By: Denis Lunev <den@sw.ru>
> Signed-Off-By: Kirill Korotaev <dev@openvz.org>
>
> --- ./mm/page_alloc.c.oomx 2006-11-08 17:44:16.000000000 +0300
> +++ ./mm/page_alloc.c 2006-11-13 21:57:33.000000000 +0300
> @@ -1251,6 +1251,7 @@ restart:
>
>   /* This allocation should allow future memory freeing. */
>
> +rebalance:
>   if (((p->flags & PF_MEMALLOC) || unlikely(test_thread_flag(TIF_MEMDIE)))
>     && !in_interrupt()) {
>    if (!(gfp_mask & __GFP_NOMEMALLOC)) {
> @@ -1272,7 +1273,6 @@ nofail_alloc:
>   if (!wait)
>    goto nopage;
>
> -rebalance:
>   cond_resched();
>
>   /* We now go into synchronous reclaim */
```

Your patch reverts a change made by Nick's
a457c255ae59b5f7f52f63fc88d5e530101772c6 two years ago.

It looks right to me, but the original change was unchangelogged and I
wonder what it was aiming to do?