
Subject: Re: [PATCH 8/8] Per-container pages reclamation
Posted by [Balbir Singh](#) on Thu, 17 May 2007 11:31:26 GMT
[View Forum Message](#) <> [Reply to Message](#)

Pavel Emelianov wrote:

- > Implement `try_to_free_pages_in_container()` to free the
- > pages in container that has run out of memory.
- >
- > The `scan_control->isolate_pages()` function isolates the
- > container pages only.
- >
- >

Hi, Pavel/Andrew,

I've started running some basic tests like Imbench and LTP vm stress on the RSS controller.

With the controller `rss_limit` set to 256 MB, I saw the following panic on a machine

Unable to handle kernel NULL pointer dereference at 000000000000001c RIP:

[<ffffff80328581>] `_raw_spin_lock+0xd/0xf6`

PGD 3c841067 PUD 5d5d067 PMD 0

Oops: 0000 [1] SMP

CPU 2

Modules linked in: ipv6 hidp rfcomm l2cap bluetooth sunrpc video button battery asus_acpi backlight ac lp parport_pc parport nvram pcspkr amd_rng rng_core i2c_amd756 i2c_core

Pid: 13581, comm: mtest01 Not tainted 2.6.20-autokern1 #1

RIP: 0010:[<ffffff80328581>] [`<ffffff80328581>`] `_raw_spin_lock+0xd/0xf6`

RSP: 0000:ffff81003e6c9ce8 EFLAGS: 00010096

RAX: ffffffff8087f720 RBX: 0000000000000018 RCX: ffff81003f36f9d0

RDX: ffff8100807bb040 RSI: 0000000000000001 RDI: 0000000000000018

RBP: 0000000000000000 R08: ffff81003e6c8000 R09: 0000000000000002

R10: ffff810001021da8 R11: ffffffff8044658f R12: ffff81000c861e01

R13: 0000000000000018 R14: ffff81000c861eb8 R15: ffff810032d34138

FS: 00002abf7a1961e0(0000) GS:ffff81003edb94c0(0000) knlGS:0000000000000000

CS: 0010 DS: 0000 ES: 0000 CR0: 000000008005003b

CR2: 000000000000001c CR3: 000000002ba6e000 CR4: 000000000000006e0

Process mtest01 (pid: 13581, threadinfo ffff81003e6c8000, task ffff81003d8ec040)

Stack: ffff810001003638 ffff810014a8c2c0 0000000000000000 ffff81000c861e01

0000000000000018 ffffffff80287166 ffff81000c861eb8 ffff81000000bac0

ffff81003f36f9a0 ffff81000c861e40 ffff81001d4b6a20 ffffffff8026a92e

Call Trace:

[<ffffff80287166>] `container_rss_move_lists+0x3b/0xaf`

[<ffffff8026a92e>] `activate_page+0xc1/0xd0`

[<ffffff80245f15>] `wake_bit_function+0x0/0x23`

[<ffffff8026ab34>] `mark_page_accessed+0x1b/0x2f`

```
[<ffffff80265d25>] filemap_nopage+0x180/0x338  
[<ffffff80270474>] __handle_mm_fault+0x1f2/0xa81  
[<ffffff804c58ef>] do_page_fault+0x42b/0x7b3  
[<ffffff802484c4>] hrtimer_cancel+0xc/0x16  
[<ffffff804c2a89>] do_nanosleep+0x47/0x70  
[<ffffff802485f4>] hrtimer_nanosleep+0x58/0x119  
[<ffffff8023bc1f>] sys_sysinfo+0x15b/0x173  
[<ffffff804c3d3d>] error_exit+0x0/0x84
```

On analyzing the code, I found that the page is mapped (we have a `page_mapped()` check in `container_rss_move_lists()`), but the `page_container` is invalid. Please review the fix attached (we reset the page's container pointer to NULL when a page is completely unmapped)

--

Warm Regards,
Balbir Singh
Linux Technology Center
IBM, ISTL
