
Subject: Re: [RFC][v2][patch 0/12][CFQ-cgroup]Yet another I/O bandwidth controlling subsystem for CGroups bas

Posted by [Andrea Righi](#) on Sun, 22 Jun 2008 17:04:33 GMT

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

Satoshi UCHIDA wrote:

> Hi, Andrea.
>
> Thanks for bug reports.
> I fix this problem.
>
> This problem causes by miss of trace for children groups.
> Please adopt and test this patch.

OK, I can confirm the fix resolves the problem.

Tested-by: Andrea Righi <righi.andrea@gmail.com>

>
> If OK, this amendment is adopted when I release new patch-set.
>
>
> Regards,
> Satoshi Uchida.
>
>
> diff --git a/block/cfq-cgroup.c b/block/cfq-cgroup.c
> index f868f4f..64561f5 100644
> --- a/block/cfq-cgroup.c
> +++ b/block/cfq-cgroup.c
> @@ -183,7 +184,7 @@ static void *cfq_cgroup_init_cfq_data(struct cfq_cgroup *cfqc, struct
cfq_data *
>
> /* check and create cfq_data for children */
> if (cfqc->css.cgroup)
> - list_for_each_entry(child, &cfqc->css.cgroup->children, children){
> + list_for_each_entry(child, &cfqc->css.cgroup->children, sibling){
> cfq_cgroup_init_cfq_data(cgroup_to_cfq_cgroup(child), cfqd);
> }
>
>
>> -----Original Message-----
>> From: Andrea Righi [mailto:righi.andrea@gmail.com]
>> Sent: Friday, May 30, 2008 7:37 PM
>> To: Satoshi UCHIDA
>> Cc: 'Ryo Tsuruta'; axboe@kernel.dk; vtaras@openvz.org;
>> containers@lists.linux-foundation.org; tom-sugawara@ap.jp.nec.com;
>> linux-kernel@vger.kernel.org

>> Subject: Re: [RFC][v2][patch 0/12][CFQ-cgroup]Yet another I/O bandwidth
>> controlling subsystem for CGroups based on CFQ
>>
>> Satoshi UCHIDA wrote:
>>> Hi, Tsuruta-san.
>>
>>> I'm looking forward to your report.
>> Hi Satoshi,
>>
>> I'm testing your patch agains latest Linus git and I've got the
>> following bug. It can be easily reproduced creating a cgroup, switching
>> the i/o scheduler from cfq to any other and switch back to cfq again.
>>
>> -Andrea
>>
>> BUG: unable to handle kernel paging request at ffffffeb
>> IP: [<c0212dc6>] cfq_cgroup_sibling_tree_add+0x36/0x90
>> Oops: 0000 [#1] SMP
>> Modules linked in: i2c_piix4 ne2k_pci 8390 i2c_core
>>
>> Pid: 3543, comm: bash Not tainted (2.6.26-rc4 #1)
>> EIP: 0060:[<c0212dc6>] EFLAGS: 00010286 CPU: 0
>> EIP is at cfq_cgroup_sibling_tree_add+0x36/0x90
>> EAX: 00000003 EBX: c7704c90 ECX: ffffffff EDX: c7102180
>> ESI: c7102240 EDI: c7704c80 EBP: c7afbe94 ESP: c7afbe80
>> DS: 007b ES: 007b FS: 00d8 GS: 0033 SS: 0068
>> Process bash (pid: 3543, ti=c7afa000 task=c7aeda00 task.ti=c7afa000)
>> Stack: c7704c90 c71022d0 c7ace078 c7704c80 c71020c0 c7afbea8 c021363c
>> c7ace078
>> c7803460 c71020c0 c7afbebc c021360a c7102184 c71020c0 c7102180
>> c7afbee4
>> c02134e0 00000000 c7102184 c72b8ab0 00000001 c7102140 c72b8ab0
>> c04b8ac0
>> Call Trace:
>> [<c021363c>] ? cfq_cgroup_init_cfq_data+0x7c/0x80
>> [<c021360a>] ? cfq_cgroup_init_cfq_data+0x4a/0x80
>> [<c02134e0>] ? __cfq_cgroup_init_queue+0x100/0x1e0
>> [<c021097b>] ? cfq_init_queue+0xb/0x10
>> [<c0204ff8>] ? elevator_init_queue+0x8/0x10
>> [<c0205cd0>] ? elv_iosched_store+0x80/0x2b0
>> [<c0209379>] ? queue_attr_store+0x49/0x70
>> [<c01c488b>] ? sysfs_write_file+0xbb/0x110
>> [<c0186276>] ? vfs_write+0x96/0x160
>> [<c01c47d0>] ? sysfs_write_file+0x0/0x110
>> [<c018696d>] ? sys_write+0x3d/0x70
>> [<c0104267>] ? sysenter_past_esp+0x78/0xd1
>> =====
>> Code: ec 08 8b 82 90 00 00 00 83 e0 fc 89 45 f0 8d 82 90 00 00 00 39 45

```
>> f0 75 5f
>> 8d 47 10 89 45 ec 89 c3 31 c0 eb 11 8b 56 7c 8d 41 04 <3b> 51 ec 8d 59
>> 08 0f 43
>> d8 89 c8 8b 0b 85 c9 75 e9 89 86 90 00
>> EIP: [<c0212dc6>] cfq_cgroup_sibling_tree_add+0x36/0x90 SS:ESP
>> 0068:c7afbe80
>> ---[ end trace 9701f4859bb53d27 ]---
```

Containers mailing list

Containers@lists.linux-foundation.org

<https://lists.linux-foundation.org/mailman/listinfo/containers>
