

---

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

Posted by [Satoshi UCHIDA](#) on Thu, 26 Jun 2008 04:49:20 GMT

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

---

Hi, Tsuruta.

> In addition, I got the following message during test #2. Program  
> "ioload", our benchmark program, was blocked more than 120 seconds.  
> Do you see any problems?

No.

I tried to test in environment which runs from 1 to 200 processes per group.

However, such message was not output.

> The result of test #1 is close to your estimation, but the result  
> of test #2 is not, the gap between the estimation and the result  
> increased.

In the above my test, the gap between the estimation and the result is increasing as a process increases.

And, in native CFQ with ionice command, this situation is a similar. These circumstances are shown in the case of more than processes of total 200.

I'll investigate this problem continuously.

Thanks,  
Satoshi Uchida.

> -----Original Message-----

> From: Ryo Tsuruta [mailto:[ryov@valinux.co.jp](mailto:ryov@valinux.co.jp)]

> Sent: Tuesday, June 03, 2008 5:16 PM

> To: [s-uchida@ap.jp.nec.com](mailto:s-uchida@ap.jp.nec.com)

> Cc: [axboe@kernel.dk](mailto:axboe@kernel.dk); [vtaras@openvz.org](mailto:vtaras@openvz.org);

> [containers@lists.linux-foundation.org](mailto:containers@lists.linux-foundation.org); [tom-sugawara@ap.jp.nec.com](mailto:tom-sugawara@ap.jp.nec.com);

> [linux-kernel@vger.kernel.org](mailto: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

>

> Hi Uchida-san,

>

> > I report my tests.

>

> I did a similar test to yours. I increased the number of I/Os

> which are issued simultaneously up to 100 per cgroup.

```

>
> Procedures:
>   o Prepare 300 files which size is 250MB on 1 partition sdb3
>   o Create three groups with priority 0, 4 and 7.
>   o Run many processes issuing random direct I/O with 4KB data on each
>     files in three groups.
>     #1 Run 25 processes issuing read I/O only per group.
>     #2 Run 100 processes issuing read I/O only per group.
>   o Count up the number of I/Os which have done in 10 minutes.
>
>     The number of I/Os (percentage to total I/O)
>
> -----
> | group   | group 1 | group 2 | group 3 | total |
> | priority | 0(highest) | 4   | 7(lowest) | I/Os |
> |-----+-----+-----+-----+-----|
> | Estimate |         |         |         |         |
> | Performance | 61.5% | 30.8% | 7.7% |         |
> |-----+-----+-----+-----+-----|
> | #1 25procs | 52763(57%) | 30811(33%) | 9575(10%) | 93149 |
> | #2 100procs | 24949(40%) | 21325(34%) | 16508(26%) | 62782 |
> |-----+-----+-----+-----+-----|
>
>
> The result of test #1 is close to your estimation, but the result
> of test #2 is not, the gap between the estimation and the result
> increased.
>
> In addition, I got the following message during test #2. Program
> "ioload", our benchmark program, was blocked more than 120 seconds.
> Do you see any problems?
>
> INFO: task ioload:8456 blocked for more than 120 seconds.
> "echo 0 > /proc/sys/kernel/hung_task_timeout_secs" disables this message.
> ioload   D 00000008 2772 8456 8419
>   f72eb740 00200082 c34862c0 00000008 c3565170 c35653c0 c2009d80
>   00000001
>   c1d1bea0 00200046 ffffffff f6ee039c 00000000 00000000 00000000
>   c2009d80
>   018db000 00000000 f71a6a00 c0604fb6 00000000 f71a6bc8 c04876a4
>   00000000
> Call Trace:
> [<c0604fb6>] io_schedule+0x4a/0x81
> [<c04876a4>] __blockdev_direct_IO+0xa04/0xb54
> [<c04a3aa2>] ext2_direct_IO+0x35/0x3a
> [<c04a4757>] ext2_get_block+0x0/0x603
> [<c044ab81>] generic_file_direct_IO+0x103/0x118
> [<c044abe6>] generic_file_direct_write+0x50/0x13d
> [<c044b59e>] __generic_file_aio_write_nolock+0x375/0x4c3
> [<c046e571>] link_path_walk+0x86/0x8f

```

> [<c044a1e8>] find\_lock\_page+0x19/0x6d  
> [<c044b73e>] generic\_file\_aio\_write+0x52/0xa9  
> [<c0466256>] do\_sync\_write+0xbf/0x100  
> [<c042ca44>] autoremove\_wake\_function+0x0/0x2d  
> [<c0413366>] update\_curr+0x83/0x116  
> [<c0605280>] mutex\_lock+0xb/0x1a  
> [<c04b653b>] security\_file\_permission+0xc/0xd  
> [<c0466197>] do\_sync\_write+0x0/0x100  
> [<c046695d>] vfs\_write+0x83/0xf6  
> [<c0466ea9>] sys\_write+0x3c/0x63  
> [<c04038de>] syscall\_call+0x7/0xb  
> [<c0600000>] print\_cpu\_info+0x27/0x92  
> =====  
>  
> Thanks,  
> Ryo Tsuruta

---

Containers mailing list  
Containers@lists.linux-foundation.org  
<https://lists.linux-foundation.org/mailman/listinfo/containers>

---