

---

Subject: [PATCH O/4] BIO tracking take2

Posted by [Hirokazu Takahashi](#) on Tue, 20 May 2008 11:59:49 GMT

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

---

Hi all,

With this series of patches, you can determine the owners of any type of I/Os. I ported the previous version to linux-2.6.26-rc2-mm1.

This makes dm-ioband -- I/O bandwidth controller -- be able to control the Block I/O bandwidths even when it accepts delayed write requests. Dm-ioband can find the owner cgroup of each request.

It is also possible that OpenVz team and NEC Uchida-san team working on the CFQ scheduler use this functionality to control asynchronous I/Os with a little enhancement.

You have to apply the patch dm-ioband v1.0.0 before applying this series of patches, which can be found at:

<http://people.valinux.co.jp/~ryov/dm-ioband>

And you have to select the following config options when compiling kernel:

CONFIG\_CGROUPS=y

CONFIG\_CGROUP\_BIO=y

And I recommend you should also select the options for cgroup memory subsystem, because it makes it possible to give some I/O bandwidth and some memory to a certain cgroup to control delayed write requests and the processes in the cgroup will be able to make pages dirty only inside the cgroup even when the given bandwidth is narrow.

CONFIG\_RESOURCE\_COUNTERS=y

CONFIG\_CGROUP\_MEM\_RES\_CTLR=y

This code is based on some part of the memory subsystem of cgroup and I think the accuracy and overhead of the subsystem can't be ignored at this time, so we need to keep tuning it up.

-----  
The following shows how to use dm-ioband with cgroups.

Please assume that you want make two cgroups, which we call "bio cgroup" here, to track down block I/Os and assign them to ioband device "ioband1".

First, mount the bio cgroup filesystem.

```
# mount -t cgroup -o bio none /cgroup/bio
```

Then, make new bio cgroups and put some processes in them.

```
# mkdir /cgroup/bio/bgroup1
```

```
# mkdir /cgroup/bio/bgroup2
# echo 1234 > /cgroup/bio/bgroup1/tasks
# echo 5678 > /cgroup/bio/bgroup1/tasks
```

Now, check the ID of each bio cgroup which is just created.

```
# cat /cgroup/bio/bgroup1/bio.id
1
# cat /cgroup/bio/bgroup2/bio.id
2
```

Finally, attach the cgroups to "ioband1" and assign them weights.

```
# dmsetup message ioband1 0 type cgroup
# dmsetup message ioband1 0 attach 1
# dmsetup message ioband1 0 attach 2
# dmsetup message ioband1 0 weight 1:30
# dmsetup message ioband1 0 weight 2:60
```

You can also make use of the dm-ioband administration tool if you want.

The tool will be found here:

<http://people.valinux.co.jp/~kaizuka/dm-ioband/iobandctl/manual.html>

You can set up the device with the tool as follows.

In this case, you don't need to know the IDs of the cgroups.

```
# iobandctl.py group /dev/mapper/ioband1 cgroup /cgroup/bio/bgroup1:30 /cgroup/bio/bgroup2:60
```

Thank you,  
Hirokazu Takahashi.

---

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

---