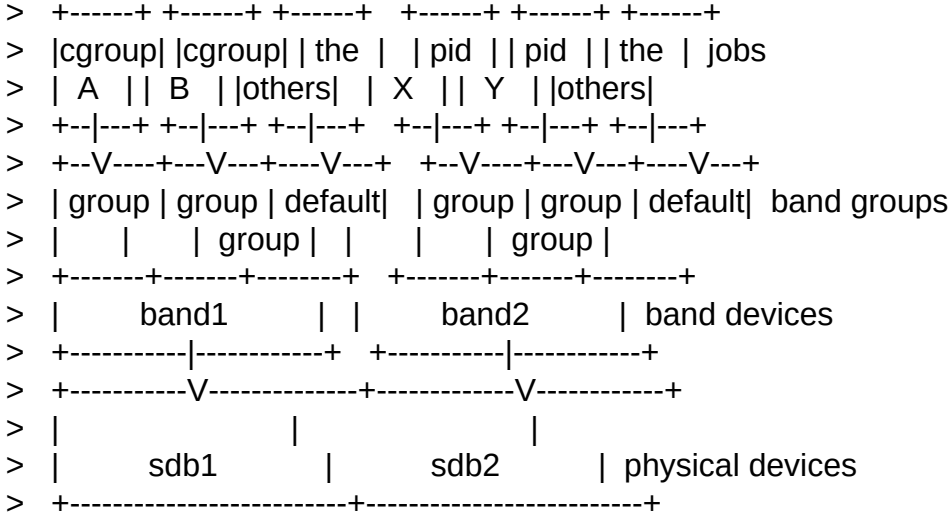
Subject: Re: [PATCH 0/2] dm-band: The I/O bandwidth controller: Overview
Posted by Anthony Liguori on Wed, 23 Jan 2008 19:22:36 GMT
View Forum Message <> Reply to Message

Hi,

I believe this work is very important especially in the context of
virtual machines. I think it would be more useful though implemented in
the context of the IO scheduler. Since we already support a notion of
IO priority, it seems reasonable to add a notion of an IO cap.

Regards,

Anthony Liguori

Ryo Tsuruta wrote:
> Hi everyone,
>
> I'm happy to announce that I've implemented a Block I/O bandwidth controller.
> The controller is designed to be of use in a cgroup or virtual machine
> environment. The current approach is that the controller is implemented as
> a device-mapper driver.
>
> What's dm-band all about?
> ========================
> Dm-band is an I/O bandwidth controller implemented as a device-mapper driver.
> Several jobs using the same physical device have to share the bandwidth of
> the device. Dm-band gives bandwidth to each job according to its weight,
> which each job can set its own value to.
>
> At this time, a job is a group of processes with the same pid or pgrp or uid.
> There is also a plan to make it support cgroup. A job can also be a virtual
> machine such as KVM or Xen.
>


>
>
> How dm-band works.
> ========================
> Every band device has one band group, which by default is called the default
> group.
>
> Band devices can also have extra band groups in them. Each band group
> has a job to support and a weight. Proportional to the weight, dm-band gives
> tokens to the group.
>
> A group passes on I/O requests that its job issues to the underlying
> layer so long as it has tokens left, while requests are blocked
> if there aren't any tokens left in the group. One token is consumed each
> time the group passes on a request. Dm-band will refill groups with tokens
> once all of groups that have requests on a given physical device use up their
> tokens.
>
> With this approach, a job running on a band group with large weight is
> guaranteed to be able to issue a large number of I/O requests.
>
>
> Getting started
> ==============
> The following is a brief description how to control the I/O bandwidth of
> disks. In this description, we'll take one disk with two partitions as an
> example target.
>
> You can also check the manual at Document/device-mapper/band.txt of the
> linux kernel source tree for more information.
>
>
> Create and map band devices
> ---------------------------
> Create two band devices "band1" and "band2" and map them to "/dev/sda1"
> and "/dev/sda2" respectively.
>
> # echo "0 `blockdev --getsize /dev/sda1` band /dev/sda1 1" | dmsetup create band1
> # echo "0 `blockdev --getsize /dev/sda2` band /dev/sda2 1" | dmsetup create band2
>
> If the commands are successful then the device files "/dev/mapper/band1"
> and "/dev/mapper/band2" will have been created.
>
>
> Bandwidth control
> -----------------
> In this example weights of 40 and 10 will be assigned to "band1" and
> "band2" respectively. This is done using the following commands:

>
> # dmsetup message band1 0 weight 40
> # dmsetup message band2 0 weight 10
>
> After these commands, "band1" can use 80% --- 40/(40+10)*100 --- of the
> bandwidth of the physical disk "/dev/sda" while "band2" can use 20%.
>
>
> Additional bandwidth control
> ---------------------------
> In this example two extra band groups are created on "band1".
> The first group consists of all the processes with user-id 1000 and the
> second group consists of all the processes with user-id 2000. Their
> weights are 30 and 20 respectively.
>
> Firstly the band group type of "band1" is set to "user".
> Then, the user-id 1000 and 2000 groups are attached to "band1".
> Finally, weights are assigned to the user-id 1000 and 2000 groups.
>
> # dmsetup message band1 0 type user
> # dmsetup message band1 0 attach 1000
> # dmsetup message band1 0 attach 2000
> # dmsetup message band1 0 weight 1000:30
> # dmsetup message band1 0 weight 2000:20
>
> Now the processes in the user-id 1000 group can use 30% ---
> 30/(30+20+40+10)*100 --- of the bandwidth of the physical disk.
>
> Band Device   Band Group   Weight
> band1   user id 1000   30
> band1   user id 2000   20
> band1   default group(the other users)   40
> band2   default group   10
>
>
> Remove band devices
> -------------------
> Remove the band devices when no longer used.
>
> # dmsetup remove band1
> # dmsetup remove band2
>
>
> TODO
> ========================
> - Cgroup support.
> - Control read and write requests separately.
> - Support WRITE_BARRIER.

> - Optimization.
> - More configuration tools. Or is the dmsetup command sufficient?
> - Other policies to schedule BIOs. Or is the weight policy sufficient?
>
> Thanks,
> Ryo Tsuruta

_____

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