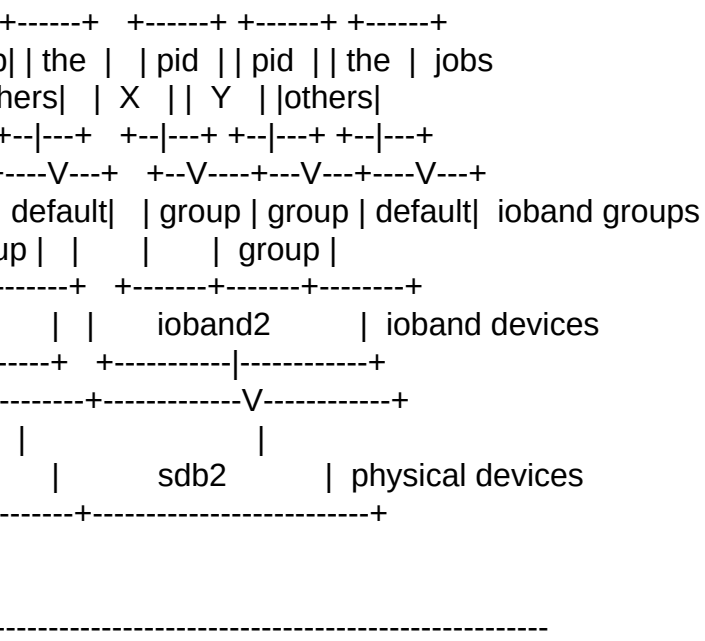
Subject: [PATCH 2/2] dm-ioband: I/O bandwidth controller v1.2.0: Document
Posted by Ryo Tsuruta on Fri, 04 Jul 2008 03:42:42 GMT
View Forum Message <> Reply to Message

Here is the document of dm-ioband.

Based on 2.6.26-rc5-mm3
Signed-off-by: Ryo Tsuruta <ryov@valinux.co.jp>
Signed-off-by: Hirokazu Takahashi <taka@valinux.co.jp>

diff -uprN linux-2.6.26-rc5-mm3.orig/Documentation/device-mapper/ioband.txt
linux-2.6.26-rc5-mm3/Documentation/device-mapper/ioband.txt
--- linux-2.6.26-rc5-mm3.orig/Documentation/device-mapper/ioband.txt 1970-01-01
09:00:00.000000000 +0900
+++ linux-2.6.26-rc5-mm3/Documentation/device-mapper/ioband.txt 2008-07-04
12:21:10.000000000 +0900
@@ -0,0 +1,933 @@
+ Block I/O bandwidth control: dm-ioband
+
+ --
+
+ Table of Contents
+
+ [1]What's dm-ioband all about?
+
+ [2]Differences from the CFQ I/O scheduler
+
+ [3]How dm-ioband works.
+
+ [4]Setup and Installation
+
+ [5]Getting started
+
+ [6]Command Reference
+
+ [7]Examples
+
+What's dm-ioband all about?
+
+ dm-ioband 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-ioband 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.
+



Differences from the CFQ I/O scheduler

Dm-ioband is flexible to configure the bandwidth settings.

Dm-ioband can work with any type of I/O scheduler such as the NOOP scheduler, which is often chosen for high-end storages, since it is implemented outside the I/O scheduling layer. It allows both of partition based bandwidth control and job --- a group of processes --- based control. In addition, it can set different configuration on each physical device to control its bandwidth.

Meanwhile the current implementation of the CFQ scheduler has 8 IO priority levels and all jobs whose processes have the same IO priority share the bandwidth assigned to this level between them. And IO priority is an attribute of a process so that it equally effects to all block devices.

How dm-ioband works.

Every ioband device has one ioband group, which by default is called the default group.

Ioband devices can also have extra ioband groups in them. Each ioband group has a job to support and a weight. Proportional to the weight, dm-ioband 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. Tokens are refilled once all of
+ groups that have requests on a given physical device use up their tokens.
+
+ There are two policies for token consumption. One is that a token is
+ consumed for each I/O request. The other is that a token is consumed for
+ each I/O sector, for example, one I/O request which consists of
+ 4Kbytes(512bytes * 8 sectors) read consumes 8 tokens. A user can choose
+ either policy.
+
+ With this approach, a job running on an ioband group with large weight
+ is guaranteed a wide I/O bandwidth.
+
+ ---
+
+Setup and Installation
+
+ Build a kernel with these options enabled:
+
+ CONFIG_MD
+ CONFIG_BLK_DEV_DM
+ CONFIG_DM_IOBAND
+
+
+ If compiled as module, use modprobe to load dm-ioband.
+
+ # make modules
+ # make modules_install
+ # depmod -a
+ # modprobe dm-ioband
+
+
+ "dmsetup targets" command shows all available device-mapper targets.
+ "ioband" is displayed if dm-ioband has been loaded.
+
+ # dmsetup targets
+ ioband v1.2.0
+
+
+ ---
+
+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.
+
+ ---
+

+ Create and map ioband devices
+
+ Create two ioband devices "ioband1" and "ioband2". "ioband1" is mapped
+ to "/dev/sda1" and has a weight of 40. "ioband2" is mapped to "/dev/sda2"
+ and has a weight of 10. "ioband1" can use 80% --- 40/(40+10)*100 --- of
+ the bandwidth of the physical disk "/dev/sda" while "ioband2" can use 20%.
+
+ # echo "0 $(blockdev --getsize /dev/sda1) ioband /dev/sda1 1 0 0 none" \
+ "weight 0 :40" | dmsetup create ioband1
+ # echo "0 $(blockdev --getsize /dev/sda2) ioband /dev/sda2 1 0 0 none" \
+ "weight 0 :10" | dmsetup create ioband2
+
+
+ If the commands are successful then the device files
+ "/dev/mapper/ioband1" and "/dev/mapper/ioband2" will have been created.
+
+ ---
+
+ Additional bandwidth control
+
+ In this example two extra ioband groups are created on "ioband1". 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.
+
+ # dmsetup message ioband1 0 type user
+ # dmsetup message ioband1 0 attach 1000
+ # dmsetup message ioband1 0 attach 2000
+ # dmsetup message ioband1 0 weight 1000:30
+ # dmsetup message ioband1 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.
+
+ Table 1. Weight assignments
+
+ +---+
+ | ioband device | ioband group | ioband weight |
+ |---------------+-------------------------------+---------------|
+ | ioband1 | user id 1000 | 30 |
+ |---------------+-------------------------------+---------------|
+ | ioband1 | user id 2000 | 20 |
+ |---------------+-------------------------------+---------------|
+ | ioband1 | default group(the other users) | 40 |
+ |---------------+-------------------------------+---------------|
+ | ioband2 | default group | 10 |
+ +---+

+
+ ---
+
+ Remove the ioband devices
+
+   Remove the ioband devices when no longer used.
+
+   # dmsetup remove ioband1
+   # dmsetup remove ioband2
+
+
+ ---
+
+Command Reference
+
+ Create an ioband device
+
+ SYNOPSIS
+
+   dmsetup create IOBAND_DEVICE
+
+ DESCRIPTION
+
+   Create an ioband device with the given name IOBAND_DEVICE.
+   Generally, dmsetup reads a table from standard input. Each line of
+   the table specifies a single target and is of the form:
+
+   start_sector num_sectors "ioband" device_file ioband_device_id \
+   io_throttle io_limit ioband_group_type policy token_base \
+   :weight [ioband_group_id:weight...]
+
+
+   start_sector, num_sectors
+
+   The sector range of the underlying device where
+   dm-ioband maps.
+
+   ioband
+
+   Specify the string "ioband" as a target type.
+
+   device_file
+
+   Underlying device name.
+
+   ioband_device_id
+
+   The ID number for an ioband device. The same ID

+ must be set among the ioband devices that share the
+ same bandwidth, which means they work on the same
+ physical disk.
+
+ io_throttle
+
+ Dm-ioband starts to control the bandwidth when the
+ number of BIOs in progress exceeds this value. If 0
+ is specified, dm-ioband uses the default value.
+
+ io_limit
+
+ Dm-ioband blocks all I/O requests for the
+ IOBAND_DEVICE when the number of BIOs in progress
+ exceeds this value. If 0 is specified, dm-ioband uses
+ the default value.
+
+ ioband_group_type
+
+ Specify how to evaluate the ioband group ID. The
+ type must be one of "none", "user", "gid", "pid" or
+ "pgrp." Specify "none" if you don't need any ioband
+ groups other than the default ioband group.
+
+ policy
+
+ Specify bandwidth control policy. A user can choose
+ either policy "weight" or "weight-iosize."
+
+ weight
+
+ This policy controls bandwidth
+ according to the proportional to the
+ weight of each ioband group based on the
+ number of I/O requests.
+
+ weight-iosize
+
+ This policy controls bandwidth
+ according to the proportional to the
+ weight of each ioband group based on the
+ number of I/O sectors.
+
+ token_base
+
+ The number of tokens which specified by token_base
+ will be distributed to all ioband groups according to
+ the proportional to the weight of each ioband group.

+ If 0 is specified, dm-ioband uses the default value.
+
+ ioband_group_id:weight
+
+ Set the weight of the ioband group specified by
+ ioband_group_id. If ioband_group_id is omitted, the
+ weight is assigned to the default ioband group.
+
+ EXAMPLE
+
+ Create an ioband device with the following parameters:
+
+ * Starting sector = "0"
+
+ * The number of sectors = "$(blockdev --getsize /dev/sda1)"
+
+ * Target type = "ioband"
+
+ * Underlying device name = "/dev/sda1"
+
+ * Ioband device ID = "128"
+
+ * I/O throttle = "10"
+
+ * I/O limit = "400"
+
+ * Ioband group type = "user"
+
+ * Bandwidth control policy = "weight"
+
+ * Token base = "2048"
+
+ * Weight for the default ioband group = "100"
+
+ * Weight for the ioband group 1000 = "80"
+
+ * Weight for the ioband group 2000 = "20"
+
+ * Ioband device name = "ioband1"
+
+ # echo "0 $(blockdev --getsize /dev/sda1) ioband" \
+ "/dev/sda1 128 10 400 user weight 2048 :100 1000:80 2000:20" \
+ | dmsetup create ioband1
+
+
+ Create two device groups (ID=1,2). The bandwidths of these
+ device groups will be individually controlled.
+

+ # echo "0 $(blockdev --getsize /dev/sda1) ioband /dev/sda1 1" \
+ "0 0 none weight 0 :80" | dmsetup create ioband1
+ # echo "0 $(blockdev --getsize /dev/sda2) ioband /dev/sda2 1" \
+ "0 0 none weight 0 :20" | dmsetup create ioband2
+ # echo "0 $(blockdev --getsize /dev/sdb3) ioband /dev/sdb3 2" \
+ "0 0 none weight 0 :60" | dmsetup create ioband3
+ # echo "0 $(blockdev --getsize /dev/sdb4) ioband /dev/sdb4 2" \
+ "0 0 none weight 0 :40" | dmsetup create ioband4
+
+
+ ---
+
+ Remove the ioband device
+
+ SYNOPSIS
+
+ dmsetup remove IOBAND_DEVICE
+
+ DESCRIPTION
+
+ Remove the specified ioband device IOBAND_DEVICE. All the band
+ groups attached to the ioband device are also removed
+ automatically.
+
+ EXAMPLE
+
+ Remove ioband device "ioband1."
+
+ # dmsetup remove ioband1
+
+
+ ---
+
+ Set an ioband group type
+
+ SYNOPSIS
+
+ dmsetup message IOBAND_DEVICE 0 type TYPE
+
+ DESCRIPTION
+
+ Set the ioband group type of the specified ioband device
+ IOBAND_DEVICE. TYPE must be one of "none", "user", "gid", "pid" or
+ "pgrp." Once the type is set, new ioband groups can be created on
+ IOBAND_DEVICE.
+
+ EXAMPLE
+

+ Set the ioband group type of ioband device "ioband1" to "user."
+
+ # dmsetup message ioband1 0 type user
+
+
+ ---
+
+ Create an ioband group
+
+ SYNOPSIS
+
+ dmsetup message IOBAND_DEVICE 0 attach ID
+
+ DESCRIPTION
+
+ Create an ioband group and attach it to IOBAND_DEVICE. ID
+ specifies user-id, group-id, process-id or process-group-id
+ depending the ioband group type of IOBAND_DEVICE.
+
+ EXAMPLE
+
+ Create an ioband group which consists of all processes with
+ user-id 1000 and attach it to ioband device "ioband1."
+
+ # dmsetup message ioband1 0 type user
+ # dmsetup message ioband1 0 attach 1000
+
+
+ ---
+
+ Detach the ioband group
+
+ SYNOPSIS
+
+ dmsetup message IOBAND_DEVICE 0 detach ID
+
+ DESCRIPTION
+
+ Detach the ioband group specified by ID from ioband device
+ IOBAND_DEVICE.
+
+ EXAMPLE
+
+ Detach the ioband group with ID "2000" from ioband device
+ "ioband2."
+
+ # dmsetup message ioband2 0 detach 1000
+

+
+ ---
+
+ Set bandwidth control policy
+
+ SYNOPSIS
+
+ dmsetup message IOBAND_DEVICE 0 policy policy
+
+ DESCRIPTION
+
+ Set bandwidth control policy. This command applies to all ioband
+ devices which have the same ioband device ID as IOBAND_DEVICE. A
+ user can choose either policy "weight" or "weight-iosize."
+
+ weight
+
+ This policy controls bandwidth according to the
+ proportional to the weight of each ioband group based
+ on the number of I/O requests.
+
+ weight-iosize
+
+ This policy controls bandwidth according to the
+ proportional to the weight of each ioband group based
+ on the number of I/O sectors.
+
+ EXAMPLE
+
+ Set bandwidth control policy of ioband devices which have the
+ same ioband device ID as "ioband1" to "weight-iosize."
+
+ # dmsetup message ioband1 0 policy weight-iosize
+
+
+ ---
+
+ Set the weight of an ioband group
+
+ SYNOPSIS
+
+ dmsetup message IOBAND_DEVICE 0 weight VAL
+
+ dmsetup message IOBAND_DEVICE 0 weight ID:VAL
+
+ DESCRIPTION
+
+ Set the weight of the ioband group specified by ID. Set the

+ weight of the default ioband group of IOBAND_DEVICE if ID isn't
+ specified.
+
+ The following example means that "ioband1" can use 80% ---
+ 40/(40+10)*100 --- of the bandwidth of the physical disk while
+ "ioband2" can use 20%.
+
+ # dmsetup message ioband1 0 weight 40
+ # dmsetup message ioband2 0 weight 10
+
+
+ The following lines have the same effect as the above:
+
+ # dmsetup message ioband1 0 weight 4
+ # dmsetup message ioband2 0 weight 1
+
+
+ VAL must be an integer larger than 0. The default value, which
+ is assigned to newly created ioband groups, is 100.
+
+ EXAMPLE
+
+ Set the weight of the default ioband group of "ioband1" to 40.
+
+ # dmsetup message ioband1 0 weight 40
+
+
+ Set the weight of the ioband group of "ioband1" with ID "1000"
+ to 10.
+
+ # dmsetup message ioband1 0 weight 1000:10
+
+
+ ---
+
+ Set the number of tokens
+
+ SYNOPSIS
+
+ dmsetup message IOBAND_DEVICE 0 token VAL
+
+ DESCRIPTION
+
+ Set the number of tokens to VAL. According to their weight, this
+ number of tokens will be distributed to all the ioband groups on
+ the physical device to which ioband device IOBAND_DEVICE belongs
+ when they use up their tokens.
+

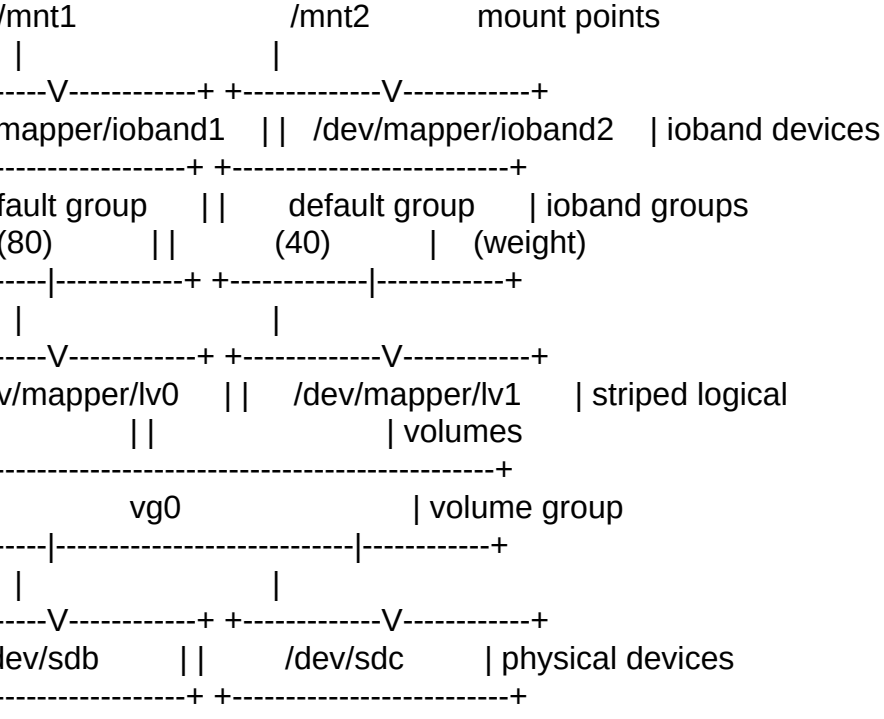
+ VAL must be an integer greater than 0. The default is 2048.
+
+ EXAMPLE
+
+ Set the number of tokens of the physical device to which
+ "ioband1" belongs to 256.
+
+ # dmsetup message ioband1 0 token 256
+
+
+ ---
+
+ Set I/O throttling
+
+ SYNOPSIS
+
+ dmsetup message IOBAND_DEVICE 0 io_throttle VAL
+
+ DESCRIPTION
+
+ Set the I/O throttling value of the physical disk to which
+ ioband device IOBAND_DEVICE belongs to VAL. Dm-ioband start to
+ control the bandwidth when the number of BIOs in progress on the
+ physical disk exceeds this value.
+
+ EXAMPLE
+
+ Set the I/O throttling value of "ioband1" to 16.
+
+ # dmsetup message ioband1 0 io_throttle 16
+
+
+ ---
+
+ Set I/O limiting
+
+ SYNOPSIS
+
+ dmsetup message IOBAND_DEVICE 0 io_limit VAL
+
+ DESCRIPTION
+
+ Set the I/O limiting value of the physical disk to which ioband
+ device IOBAND_DEVICE belongs to VAL. Dm-ioband will block all I/O
+ requests for the physical device if the number of BIOs in progress
+ on the physical disk exceeds this value.
+
+ EXAMPLE

+
+ Set the I/O limiting value of "ioband1" to 128.
+
+ # dmsetup message ioband1 0 io_limit 128
+
+
+ ---
+
+ Display settings
+
+ SYNOPSIS
+
+ dmsetup table --target ioband
+
+ DESCRIPTION
+
+ Display the current table for the ioband device in a format. See
+ "dmsetup create" command for information on the table format.
+
+ EXAMPLE
+
+ The following output shows the current table of "ioband1."
+
+ # dmsetup table --target ioband
+ ioband: 0 32129937 ioband1 8:29 128 10 400 user weight \
+ 2048 :100 1000:80 2000:20
+
+
+ ---
+
+ Display Statistics
+
+ SYNOPSIS
+
+ dmsetup status --target ioband
+
+ DESCRIPTION
+
+ Display the statistics of all the ioband devices whose target
+ type is "ioband."
+
+ The output format is as below. the first five columns shows:
+
+ * ioband device name
+
+ * logical start sector of the device (must be 0)
+
+ * device size in sectors

+
+ * target type (must be "ioband")
+
+ * device group ID
+
+ The remaining columns show the statistics of each ioband group
+ on the band device. Each group uses seven columns for its
+ statistics.
+
+ * ioband group ID (-1 means default)
+
+ * total read requests
+
+ * delayed read requests
+
+ * total read sectors
+
+ * total write requests
+
+ * delayed write requests
+
+ * total write sectors
+
+ EXAMPLE
+
+ The following output shows the statistics of two ioband devices.
+ Ioband2 only has the default ioband group and ioband1 has three
+ (default, 1001, 1002) ioband groups.
+
+ # dmsetup status
+ ioband2: 0 44371467 ioband 128 -1 143 90 424 122 78 352
+ ioband1: 0 44371467 ioband 128 -1 223 172 408 211 136 600 1001 \
+ 166 107 472 139 95 352 1002 211 146 520 210 147 504
+
+
+ ---
+
+ Reset status counter
+
+ SYNOPSIS
+
+ dmsetup message IOBAND_DEVICE 0 reset
+
+ DESCRIPTION
+
+ Reset the statistics of ioband device IOBAND_DEVICE.
+
+ EXAMPLE

+
+ Reset the statistics of "ioband1."
+
+ # dmsetup message ioband1 0 reset
+
+
+ ---
+
+Examples
+
+ Example #1: Bandwidth control on Partitions
+
+ This example describes how to control the bandwidth with disk
+ partitions. The following diagram illustrates the configuration of this
+ example. You may want to run a database on /dev/mapper/ioband1 and web
+ applications on /dev/mapper/ioband2.
+
+ /mnt1 /mnt2 mount points
+ | |
+ +----------V----------+ +----------V----------+
+ | /dev/mapper/ioband1 | | /dev/mapper/ioband2 | ioband devices
+ +---------------------+ +---------------------+
+ | default group | | default group | ioband groups
+ | (80) | | (40) | (weight)
+ +----------|----------+ +----------|----------+
+ | |
+ +----------V----------+----------V----------+
+ | /dev/sda1 | /dev/sda2 | physical devices
+ +---------------------+---------------------+
+
+
+ To setup the above configuration, follow these steps:
+
+ 1. Create ioband devices with the same device group ID and assign
+ weights of 80 and 40 to the default ioband groups respectively.
+
+ # echo "0 $(blockdev --getsize /dev/sda1) ioband /dev/sda1 1 0 0" \
+ "none weight 0 :80" | dmsetup create ioband1
+ # echo "0 $(blockdev --getsize /dev/sda2) ioband /dev/sda2 1 0 0" \
+ "none weight 0 :40" | dmsetup create ioband2
+
+
+ 2. Create filesystems on the ioband devices and mount them.
+
+ # mkfs.ext3 /dev/mapper/ioband1
+ # mount /dev/mapper/ioband1 /mnt1
+
+ # mkfs.ext3 /dev/mapper/ioband2

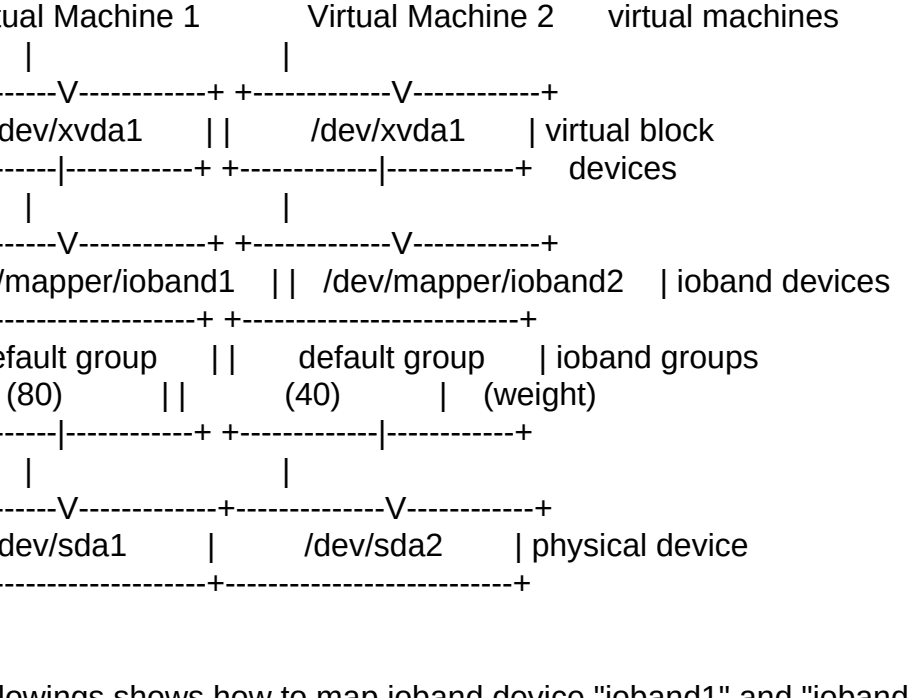
+ # mount /dev/mapper/ioband2 /mnt2
+
+
+ ---
+
+ Example #2: Bandwidth control on Logical Volumes
+
+ This example is similar to the example #1 but it uses LVM logical
+ volumes instead of disk partitions. This example shows how to configure
+ ioband devices on two striped logical volumes.
+



+
+ To setup the above configuration, follow these steps:
+
+ 1. Initialize the partitions for use by LVM.
+
+ # pvcreate /dev/sdb
+ # pvcreate /dev/sdc
+
+
+ 2. Create a new volume group named "vg0" with /dev/sdb and /dev/sdc.
+
+ # vgcreate vg0 /dev/sdb /dev/sdc
+
+
+ 3. Create two logical volumes in "vg0." The volumes have to be striped.
+
+ # lvcreate -n lv0 -i 2 -l 64 vg0 -L 1024M

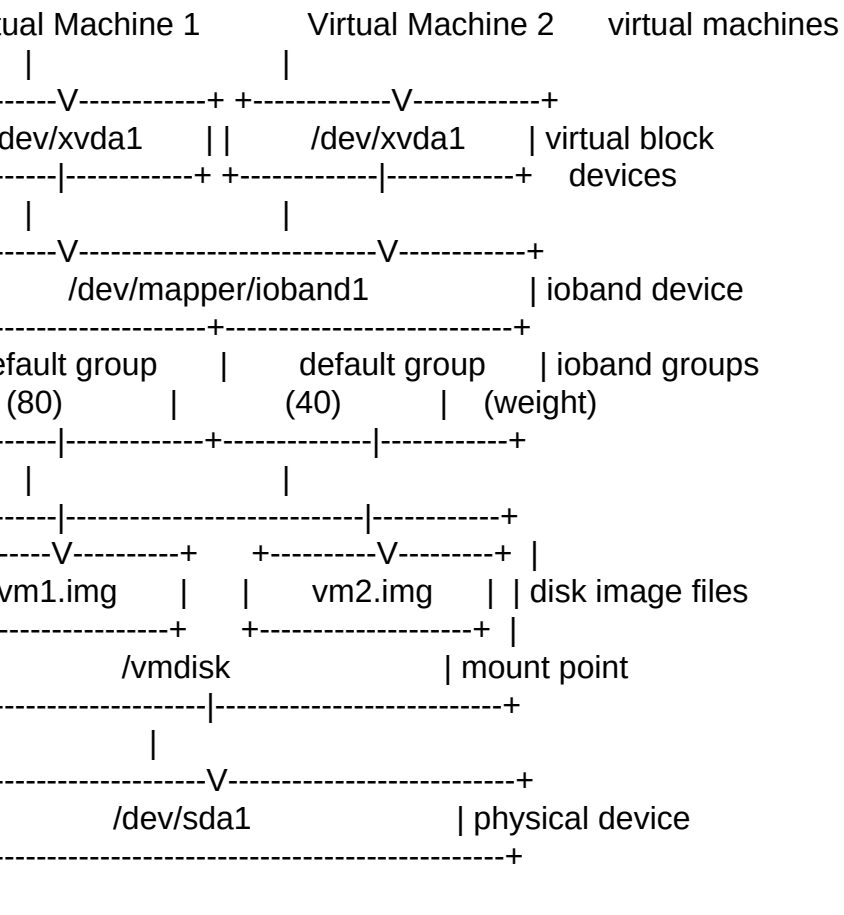
```
+       # lvcreate -n lv1 -i 2 -l 64 vg0 -L 1024M
+
+
+       The rest is the same as the example #1.
+
+   4.  Create ioband devices corresponding to each logical volume and
+     assign weights of 80 and 40 to the default ioband groups respectively.
+
+       # echo "0 $(blockdev --getsize /dev/mapper/vg0-lv0)" \
+         "ioband /dev/mapper/vg0-lv0 1 0 0 none weight 0 :80" | \
+         dmsetup create ioband1
+       # echo "0 $(blockdev --getsize /dev/mapper/vg0-lv1)" \
+         "ioband /dev/mapper/vg0-lv1 1 0 0 none weight 0 :40" | \
+         dmsetup create ioband2
+
+
+   5.  Create filesystems on the ioband devices and mount them.
+
+       # mkfs.ext3 /dev/mapper/ioband1
+       # mount /dev/mapper/ioband1 /mnt1
+
+       # mkfs.ext3 /dev/mapper/ioband2
+       # mount /dev/mapper/ioband2 /mnt2
+
+
+  ------------------------------------------------------------------------
+
+ Example #3: Bandwidth control on processes
+
+    This example describes how to control the bandwidth with groups of
+  processes. You may also want to run an additional application on the same
+  machine described in the example #1. This example shows how to add a new
+  ioband group for this application.
+
+              /mnt1                      /mnt2           mount points
+                |                          |
+    +-------------V------------+ +-------------V------------+
+    |   /dev/mapper/ioband1    | |   /dev/mapper/ioband2    | ioband devices
+    +-------------+------------+ +-------------+------------+
+    |         default          | |  user=1000  |   default  | ioband groups
+    |          (80)            | |     (20)    |    (40)    |   (weight)
+    +-------------+------------+ +-------------+------------+
+                  |                          |
+    +-------------V-------------+-------------V------------+
+    |        /dev/sda1          |          /dev/sda2       | physical device
+    +---------------------------+--------------------------+
+
+
```

+ The following shows to set up a new ioband group on the machine that is
+ already configured as the example #1. The application will have a weight
+ of 20 and run with user-id 1000 on /dev/mapper/ioband2.
+
+ 1. Set the type of ioband2 to "user."
+
+ # dmsetup message ioband2 0 type user.
+
+
+ 2. Create a new ioband group on ioband2.
+
+ # dmsetup message ioband2 0 attach 1000
+
+
+ 3. Assign weight of 10 to this newly created ioband group.
+
+ # dmsetup message ioband2 0 weight 1000:20
+
+
+ ---
+
+ Example #4: Bandwidth control for Xen virtual block devices
+
+ This example describes how to control the bandwidth for Xen virtual
+ block devices. The following diagram illustrates the configuration of this
+ example.
+



+
+ The followings shows how to map ioband device "ioband1" and "ioband2" to
+ virtual block device "/dev/xvda1 on Virtual Machine 1" and "/dev/xvda1 on
+ Virtual Machine 2" respectively on the machine configured as the example

+ #1. Add the following lines to the configuration files that are referenced
+ when creating "Virtual Machine 1" and "Virtual Machine 2."
+
+ For "Virtual Machine 1"
+ disk = ['phy:/dev/mapper/ioband1,xvda,w']
+
+ For "Virtual Machine 2"
+ disk = ['phy:/dev/mapper/ioband2,xvda,w']
+
+
+ ---
+
+ Example #5: Bandwidth control for Xen blktap devices
+
+ This example describes how to control the bandwidth for Xen virtual
+ block devices when Xen blktap devices are used. The following diagram
+ illustrates the configuration of this example.
+



+
+ To setup the above configuration, follow these steps:
+
+ 1. Create an ioband device.
+
+ # echo "0 $(blockdev --getsize /dev/sda1) ioband /dev/sda1" \

+ "1 0 0 none weight 0 :100" | dmsetup create ioband1
+
+
+ 2. Add the following lines to the configuration files that are
+ referenced when creating "Virtual Machine 1" and "Virtual Machine 2."
+ Disk image files "/vmdisk/vm1.img" and "/vmdisk/vm2.img" will be used.
+
+ For "Virtual Machine 1"
+ disk = ['tap:aio:/vmdisk/vm1.img,xvda,w',]
+
+ For "Virtual Machine 1"
+ disk = ['tap:aio:/vmdisk/vm2.img,xvda,w',]
+
+
+ 3. Run the virtual machines.
+
+ # xm create vm1
+ # xm create vm2
+
+
+ 4. Find out the process IDs of the daemons which control the blktap
+ devices.
+
+ # lsof /vmdisk/disk[12].img
+ COMMAND PID USER FD TYPE DEVICE SIZE NODE NAME
+ tapdisk 15011 root 11u REG 253,0 2147483648 48961 /vmdisk/vm1.img
+ tapdisk 15276 root 13u REG 253,0 2147483648 48962 /vmdisk/vm2.img
+
+
+ 5. Create new ioband groups of pid 15011 and pid 15276, which are
+ process IDs of the tapdisks, and assign weight of 80 and 40 to the
+ groups respectively.
+
+ # dmsetup message ioband1 0 type pid
+ # dmsetup message ioband1 0 attach 15011
+ # dmsetup message ioband1 0 weight 15011:80
+ # dmsetup message ioband1 0 attach 15276
+ # dmsetup message ioband1 0 weight 15276:40