Subject: [PATCH 2/2] dm-band: The I/O bandwidth controller: Document
Posted by Ryo Tsuruta on Wed, 23 Jan 2008 12:58:44 GMT

View Forum Message <> Reply to Message

Here is the document of dm-band.

Based on 2.6.23.14
Signed-off-by: Ryo Tsuruta <ryov@valinux.co.jp>
Signed-off-by: Hirokazu Takahashi <taka@valinux.co.jp>

diff -uprN linux-2.6.23.14.orig/Documentation/device-mapper/band.txt
linux-2.6.23.14/Documentation/device-mapper/band.txt
--- linux-2.6.23.14.orig/Documentation/device-mapper/band.txt 1970-01-01 09:00:00.000000000
+0900
+++ linux-2.6.23.14/Documentation/device-mapper/band.txt 2008-01-23 21:48:46.000000000
+0900
@@ -0,0 +1,431 @@
+====================
+Document for dm-band
+====================
+
+Contents:
+ What's dm-band all about?
+ How dm-band works
+ Setup and Installation
+ Command Reference
+ TODO
+
+
+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.
+
+ +------+ +------+ +------+ +------+ +------+ +------+
+ |cgroup| |cgroup| | the | | pid | | pid | | the | jobs
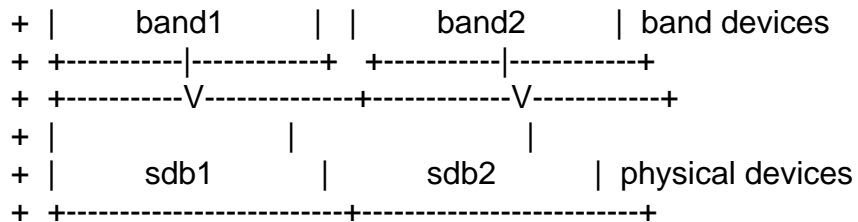+ | A || B ||others| | X || Y | |others|
+ +--|---+ +--|---+ +--|---+ +--|---+ +--|---+ +--|---+
+ +--V----+---V---+----V---+ +--V----+---V---+----V---+
+ | group | group | default| | group | group | default| band groups
+ | | | group | | | | group |
+ +-------+-------+--------+ +-------+-------+--------+

```
+ |        band1        |  |       band2        | band devices
+ +----------|-----------+  +----------|-----------+
+ +----------V--------------+------------V-----------+
+ |                     |                     |
+ |        sdb1         |        sdb2         | physical devices
+ +------------------------+------------------------+
+
+
```

+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.
+
+
+Setup and Installation
+======================
+
+Build a kernel with these options enabled:
+
+ CONFIG_MD
+ CONFIG_BLK_DEV_DM
+ CONFIG_DM_BAND
+
+If compiled as module, use modprobe to load dm-band.
+
+ # make modules
+ # make modules_install
+ # depmod -a
+ # modprobe dm-band
+
+"dmsetup targets" command shows all available device-mapper targets.
+"band" is displayed if dm-band has loaded.
+
+ # dmsetup targets

```
+ band            v0.0.2
+
+
+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 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
+
+
+Command Reference
+=================
+
+
+Create a band device
+---------------------
+SYNOPSIS
+ dmsetup create BAND_DEVICE
+
+DESCRIPTION
+ The following space delimited arguments, which describe the physical device
+ may are read from standard input. All arguments are required, and they must
+ be provided in order the order listed below.
+
+   starting sector of the physical device
+   size in sectors of the physical device
+   string "band" as a target type
+   physical device name
+   device group ID
+
+ You must set the same device group ID for each band device that shares
+ the same bandwidth.
+
+ A default band group is also created and attached to the band device.
+
+ If the command is successful, the device file
+ "/dev/device-mapper/BAND_DEVICE" will have been created.

```
+
+EXAMPLE
+  Create a band device with the following parameters:
+    physical device = "/dev/sda1"
+    band device name = "band1"
+    device group ID = "100"
+
+    # size=`blockdev --getsize /dev/sda1`
+    # echo "0 $size band /dev/sda1 100" | dmsetup create band1
+
+  Create two device groups (ID=1,2). The bandwidth of each device group may be
+  individually controlled.
+
+    # echo "0 11096883 band /dev/sda1 1" | dmsetup create band1
+    # echo "0 11096883 band /dev/sda2 1" | dmsetup create band2
+    # echo "0 11096883 band /dev/sda3 2" | dmsetup create band3
+    # echo "0 11096883 band /dev/sda4 2" | dmsetup create band4
+
+
+Remove the band device
+----------------------
+SYNOPSIS
+  dmsetup remove BAND_DEVICE
+
+DESCRIPTION
+  Remove the band device with the given name. All band groups that are attached
+  to the band device are removed automatically.
+
+EXAMPLE
+  Remove the band device "band1".
+
+  # dmsetup remove band1
+
+
+Set a band group's type
+-----------------------
+SYNOPSIS
+  dmsetup message BAND_DEVICE 0 type TYPE
+
+DESCRIPTION
+  Set a band group's type. TYPE must be one of "user", "pid" or "pgrp".
+
+EXAMPLE
+  Set a band group's type to "user".
+
+  # dmsetup message band1 0 type user
+
+
```

+Create a band group
+--------------------
+SYNOPSIS
+ dmsetup message BAND_DEVICE 0 attach ID
+
+DESCRIPTION
+ Create a band group and attach it a band device. The ID number specifies the
+ user-id, pid or pgrp, as per the the type.
+
+EXAMPLE
+ Attach a band group with uid 1000 to the band device "band1".
+
+ # dmsetup message band1 0 type user
+ # dmsetup message band1 0 attach 1000
+
+
+Remove a band group
+--------------------
+SYNOPSIS
+ dmsetup message BAND_DEVICE 0 detach ID
+
+DESCRIPTION
+ Detach a band group specified by ID from a band device.
+
+EXAMPLE
+ Detach the band group with ID "2000" from the band device "band2".
+
+ # dmsetup message band2 0 detach 1000
+
+
+Set the weight of a band group
+------------------------------
+SYNOPSIS
+ dmsetup message BAND_DEVICE 0 weight VAL
+ dmsetup message BAND_DEVICE 0 weight ID:VAL
+
+DESCRIPTION
+ Set the weight of band group. The weight is evaluated as a ratio against the
+ total weight. The following example means that "band1" can use 80% ---
+ 40/(40+10)*100 --- of the bandwidth of the physical disk "/dev/sda" while
+ "band2" can use 20%.
+
+   # dmsetup message band1 0 weight 40
+   # dmsetup message band1 0 weight 10
+
+ The following has the same effect as the above commands:
+
+   # dmsetup message band1 0 weight 4

```
+    # dmsetup message band2 0 weight 1
+
+  VAL must be an integer grater than 0. The default is 100.
+
+EXAMPLE
+  Set the weight of the default band group to 40.
+
+  # dmsetup message band1 0 weight 40
+
+  Set the weight of the band group with ID "1000" to 10.
+
+  # dmsetup message band1 0 weight 1000:10
+
+
+Set the number of tokens
+------------------------
+SYNOPSIS
+  dmsetup message BAND_DEVICE 0 token VAL
+
+DESCRIPTION
+  Set the number of tokens. The value is applied to the all band devices
+  that have the same device group ID as BAND_DEVICE.
+  VAL must be an integer grater than 0. The default is 2048.
+
+EXAMPLE
+  Set a token to 256.
+
+  # dmsetup message band1 0 token 256
+
+
+Set I/O throttling
+------------------
+SYNOPSIS
+  dmsetup message BAND_DEVICE 0 io_throttle VAL
+
+DESCRIPTION
+  Set I/O throttling. The value is applied to all band devices that have the
+  same device group ID as BAND_DEVICE.
+  VAL must be an integer grater than 0. The default is 4.
+
+  I/O requests are throttled up until the number of in-progress I/Os reaches
+  this value.
+
+EXAMPLE
+  Set I/O throttling to 16.
+
+  # dmsetup message band1 0 io_throttle 16
+
```

```
+
+Set I/O limiting
+----------------
+SYNOPSIS
+  dmsetup message BAND_DEVICE 0 io_limit VAL
+
+DESCRIPTION
+  Set I/O limiting. The value is applied to the all band devices that have
+  the same device group ID as BAND_DEVICE.
+  VAL must be an integer grater than 0. The default is 128.
+
+  When the number of in-progress I/Os reaches this value, subsequent I/O
+  requests are blocked.
+
+EXAMPLE
+  Set an io_limit to 128.
+
+  # dmsetup message band1 0 io_limit 128
+
+
+Display settings
+----------------
+SYNOPSIS
+  dmsetup table --target band
+
+DESCRIPTION
+  Display the settings of each band device.
+
+  The output format is as below:
+    On the first line for a device, space delimited.
+      Band device name
+      Starting sector of partition
+      Partition size in sectors
+      Target type
+      Device number (major:minor)
+      Device group ID
+      I/O throttle
+      I/O limit
+
+    On subsequent indented lines for a device, space delimited.
+      Group ID
+      Group type
+      Weight
+      Token
+
+EXAMPLE
+  # dmsetup table --target band
+  band2: 0 11096883 band 8:30 devgrp=0 io_throttle=4 io_limit=128
```

+   id=default type=none weight=20 token=205
+   band1: 0 11096883 band 8:31 devgrp=0 io_throttle=4 io_limit=128
+   id=default type=user weight=80 token=820
+   id=1000 weight=80 token=820
+   id=2000 weight=20 token=205
+
+
+Display Statistics
+------------------
+SYNOPSIS
+   dmsetup status --target band
+
+DESCRIPTION
+   Display the statistics of each band device.
+
+   The output format is as below:
+   On the first line for a device, space delimited.
+   Band Device Name
+   Start Sector of Device
+   Device Size in Sectors
+   Target Type
+   Device Group ID
+
+   On subsequent indented lines for a device, space delimited.
+   "parent" or Group ID,
+   Total read requests
+   Delayed read requests
+   Total read sectors
+   Total write requests
+   Delayed write requests
+   Total write sectors
+
+EXAMPLE
+   # dmsetup status
+   band2: 0 11096883 band devgrp=0 # read-req delay sect write-req delay sect
+   parent 913 898 7304 899 886 7192
+   band1: 0 11096883 band devgrp=0 # read-req delay sect write-req delay sect
+   parent 121 100 968 101 85 808
+   1000 482 468 3856 491 473 3928
+   2000 502 489 4016 469 448 3752
+
+
+Reset status counter
+--------------------
+SYNOPSIS
+   dmsetup message BAND_DEVICE 0 reset
+
+DESCRIPTION

+ Reset the status counter of a band device.
+
+EXAMPLE
+ Reset the "band1" counter.
+
+ # dmsetup message band1 0 reset
+
+
+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?

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