Block IO Controller

Overview

cgroup subsys "blkio" implements the block io controller. There seems to be a need of various kinds of IO control policies (like proportional BW, max BW) both at leaf nodes as well as at intermediate nodes in a storage hierarchy. Plan is to use the same cgroup based management interface for blkio controller and based on user options switch IO policies in the background.

One IO control policy is throttling policy which can be used to specify upper IO rate limits on devices. This policy is implemented in generic block layer and can be used on leaf nodes as well as higher level logical devices like device mapper.

HOWTO

Throttling/Upper Limit policy

Enable Block IO controller:

```
CONFIG_BLK_CGROUP=y
```

Enable throttling in block layer:

```
CONFIG_BLK_DEV_THROTTLING=y
```

Mount blkio controller (see cgroups.txt, Why are cgroups needed?):

```
mount -t cgroup -o blkio none /sys/fs/cgroup/blkio
```

Specify a bandwidth rate on particular device for root group. The format for policy is "<major>:<minor> <bytes_per_second>":

```
echo "8:16  1048576" > /sys/fs/cgroup/blkio/blkio.throttle.read_bps_device
```

This will put a limit of 1MB/second on reads happening for root group on device having major/minor number 8:16.

Run dd to read a file and see if rate is throttled to 1MB/s or not:

```
# dd iflag=direct if=/mnt/common/zerofile of=/dev/null bs=4K count=1024
1024+0 records in
1024+0 records out
4194304 bytes (4.2 MB) copied, 4.0001 s, 1.0 MB/s
```

Limits for writes can be put using blkio.throttle.write_bps_device file.

Hierarchical Cgroups

Throttling implements hierarchy support; however, throttling's hierarchy support is enabled iff "sane_behavior" is enabled from cgroup side, which currently is a development option and not publicly available.

If somebody created a hierarchy like as follows:

```
   root
   /  \
test1 test2
   |
test3
```

Throttling with "sane_behavior" will handle the hierarchy correctly. For throttling, all limits apply to the whole subtree while all statistics are local to the IOs directly generated by tasks in that cgroup.

Throttling without "sane_behavior" enabled from cgroup side will practically treat all groups at same level as if it looks like the following:

```
        pivot
     /  /   \  \
root  test1 test2  test3
```

Various user visible config options

CONFIG_BLK_CGROUP
: Block IO controller.

CONFIG_BFQ_CGROUP_DEBUG
: Debug help. Right now some additional stats file show up in cgroup if this option is enabled.

CONFIG_BLK_DEV_THROTTLING

Enable block device throttling support in block layer.

Details of cgroup files

Proportional weight policy files

blkio.bfq.weight

Specifies per cgroup weight. This is default weight of the group on all the devices until and unless overridden by per device rule (see *blkio.bfq.weight_device* below).

Currently allowed range of weights is from 1 to 1000. For more details, see Documentation/block/bfq-iosched.rst.

blkio.bfq.weight_device

Specifes per cgroup per device weights, overriding the default group weight. For more details, see Documentation/block/bfq-iosched.rst.

Following is the format:

```
# echo dev_maj:dev_minor weight > blkio.bfq.weight_device
```

Configure weight=300 on /dev/sdb (8:16) in this cgroup:

```
# echo 8:16 300 > blkio.bfq.weight_device
# cat blkio.bfq.weight_device
dev     weight
8:16    300
```

Configure weight=500 on /dev/sda (8:0) in this cgroup:

```
# echo 8:0 500 > blkio.bfq.weight_device
# cat blkio.bfq.weight_device
dev     weight
8:0     500
8:16    300
```

Remove specific weight for /dev/sda in this cgroup:

```
# echo 8:0 0 > blkio.bfq.weight_device
# cat blkio.bfq.weight_device
dev     weight
8:16    300
```

blkio.time

Disk time allocated to cgroup per device in milliseconds. First two fields specify the major and minor number of the device and third field specifies the disk time allocated to group in milliseconds.

blkio.sectors

Number of sectors transferred to/from disk by the group. First two fields specify the major and minor number of the device and third field specifies the number of sectors transferred by the group to/from the device.

blkio.io_service_bytes

Number of bytes transferred to/from the disk by the group. These are further divided by the type of operation - read or write, sync or async. First two fields specify the major and minor number of the device, third field specifies the operation type and the fourth field specifies the number of bytes.

blkio.io_serviced

Number of IOs (bio) issued to the disk by the group. These are further divided by the type of operation - read or write, sync or async. First two fields specify the major and minor number of the device, third field specifies the operation type and the fourth field specifies the number of IOs.

blkio.io_service_time

Total amount of time between request dispatch and request completion for the IOs done by this cgroup. This is in nanoseconds to make it meaningful for flash devices too. For devices with queue depth of 1, this time represents the actual service time. When queue_depth > 1, that is no longer true as requests may be served out of order. This may cause the service time for a given IO to include the service time of multiple IOs when served out of order which may result in total io_service_time > actual time elapsed. This time is further divided by the type of operation - read or write, sync or async. First two fields specify the major and minor number of the device, third field specifies the operation type and the fourth field specifies the io_service_time in ns.

blkio.io_wait_time

Total amount of time the IOs for this cgroup spent waiting in the scheduler queues for service. This can be greater than the total time elapsed since it is cumulative io_wait_time for all IOs. It is not a measure of total time the cgroup spent waiting but rather a measure of the wait_time for its individual IOs. For devices with queue_depth > 1 this metric does not include the time spent waiting for service once the IO is dispatched to the device but till it actually gets serviced (there might be a time lag here due to re-ordering of requests by the device). This is in nanoseconds to make it meaningful for flash devices too. This time is further divided by the type of operation - read or write, sync or async. First two fields specify the major and minor number of the device, third field specifies the operation type and the fourth field specifies the io_wait_time in ns.

blkio.io_merged

Total number of bios/requests merged into requests belonging to this cgroup. This is further divided by the type of operation - read or write, sync or async.

blkio.io_queued

Total number of requests queued up at any given instant for this cgroup. This is further divided by the type of operation - read or write, sync or async.

blkio.avg_queue_size

Debugging aid only enabled if CONFIG_BFQ_CGROUP_DEBUG=y. The average queue size for this cgroup over the entire time of this cgroup's existence. Queue size samples are taken each time one of the queues of this cgroup gets a timeslice.

blkio.group_wait_time

Debugging aid only enabled if CONFIG_BFQ_CGROUP_DEBUG=y. This is the amount of time the cgroup had to wait since it became busy (i.e., went from 0 to 1 request queued) to get a timeslice for one of its queues. This is different from the io_wait_time which is the cumulative total of the amount of time spent by each IO in that cgroup waiting in the scheduler queue. This is in nanoseconds. If this is read when the cgroup is in a waiting (for timeslice) state, the stat will only report the group_wait_time accumulated till the last time it got a timeslice and will not include the current delta.

blkio.empty_time

Debugging aid only enabled if CONFIG_BFQ_CGROUP_DEBUG=y. This is the amount of time a cgroup spends without any pending requests when not being served, i.e., it does not include any time spent idling for one of the queues of the cgroup. This is in nanoseconds. If this is read when the cgroup is in an empty state, the stat will only report the empty_time accumulated till the last time it had a pending request and will not include the current delta.

blkio.idle_time

Debugging aid only enabled if CONFIG_BFQ_CGROUP_DEBUG=y. This is the amount of time spent by the IO scheduler idling for a given cgroup in anticipation of a better request than the existing ones from other queues/cgroups. This is in nanoseconds. If this is read when the cgroup is in an idling state, the stat will only report the idle_time accumulated till the last idle period and will not include the current delta.

blkio.dequeue

Debugging aid only enabled if CONFIG_BFQ_CGROUP_DEBUG=y. This gives the statistics about how many a times a group was dequeued from service tree of the device. First two fields specify the major and minor number of the device and third field specifies the number of times a group was dequeued from a particular device.

blkio.*_recursive

Recursive version of various stats. These files show the same information as their non-recursive counterparts but include stats from all the descendant cgroups.

Throttling/Upper limit policy files

blkio.throttle.read_bps_device

Specifies upper limit on READ rate from the device. IO rate is specified in bytes per second. Rules are per device. Following is the format:

```
echo "<major>:<minor>  <rate_bytes_per_second>" > /cgrp/blkio.throttle.read_bps_device
```

blkio.throttle.write_bps_device

Specifies upper limit on WRITE rate to the device. IO rate is specified in bytes per second. Rules are per device. Following is the format:

```
echo "<major>:<minor>  <rate_bytes_per_second>" > /cgrp/blkio.throttle.write_bps_device
```

blkio.throttle.read_iops_device

Specifies upper limit on READ rate from the device. IO rate is specified in IO per second. Rules are per device.

Following is the format:

```
echo "<major>:<minor>  <rate_io_per_second>" > /cgrp/blkio.throttle.read_iops_device
```

blkio.throttle.write_iops_device

Specifies upper limit on WRITE rate to the device. IO rate is specified in io per second. Rules are per device. Following is the format:

```
echo "<major>:<minor>  <rate_io_per_second>" > /cgrp/blkio.throttle.write_iops_device
```

Note: If both BW and IOPS rules are specified for a device, then IO is subjected to both the constraints.

blkio.throttle.io_serviced

Number of IOs (bio) issued to the disk by the group. These are further divided by the type of operation - read or write, sync or async. First two fields specify the major and minor number of the device, third field specifies the operation type and the fourth field specifies the number of IOs.

blkio.throttle.io_service_bytes

Number of bytes transferred to/from the disk by the group. These are further divided by the type of operation - read or write, sync or async. First two fields specify the major and minor number of the device, third field specifies the operation type and the fourth field specifies the number of bytes.

Common files among various policies

blkio.reset_stats

Writing an int to this file will result in resetting all the stats for that cgroup.