

# Re: RFC: I/O bandwidth controller

---

*Source:* <http://linux.derkeiler.com/Mailing-Lists/Kernel/2008-08/msg05191.html>

---

- *From:* Andrea Righi <[righi.andrea@xxxxxxxxxx](mailto:righi.andrea@xxxxxxxxxx)>
  - *Date:* Tue, 12 Aug 2008 14:55:48 +0200 (MEST)
- 

Hirokazu Takahashi wrote:

3. & 4. & 5.  
– I/O  
bandwidth  
shaping &  
General  
design  
aspects

The  
implementation  
of an I/O  
scheduling  
algorithm is  
to a certain  
extent  
influenced  
by what we  
are trying to  
achieve in  
terms of I/O  
bandwidth  
shaping,  
but, as  
discussed  
below, the  
required  
accuracy  
can  
determine  
the layer  
where the  
I/O  
controller  
has to  
reside. Off  
the top of  
my

Re: RFC: I/O bandwidth controller

head, there  
are three  
basic  
operations  
we may  
want  
perform:  
– I/O nice  
prioritization:  
ionice-like  
approach.

–  
Proportional  
bandwidth  
scheduling:  
each  
process/group  
of processes  
has a weight  
that  
determines  
the share of  
bandwidth  
they  
receive.  
– I/O  
limiting: set  
an upper  
limit to the  
bandwidth a  
group of  
tasks  
can use.

Use a deadline-based IO  
scheduling could be an  
interesting path to be  
explored as well, IMHO, to  
try to guarantee per-cgroup  
minimum bandwidth  
requirements.

Please note that the only thing we can do is  
to guarantee minimum  
bandwidth requirement when there is  
contention for an IO resource, which  
is precisely what a proportional bandwidth  
scheduler does. Am I missing  
something?

Re: RFC: I/O bandwidth controller

Correct. Proportional bandwidth automatically allows to guarantee minimum requirements (instead of IO limiting approach, that needs additional mechanisms to achieve this).

In any case there's no guarantee for a cgroup/application to sustain i.e. 10MB/s on a certain device, but this is a hard problem anyway, and the best we can do is to try to satisfy "soft" constraints.

I think guaranteeing the minimum I/O bandwidth is very important. In the business site, especially in streaming service system, administrator requires the functionality to satisfy QoS or performance of their service. Of course, IO throttling is important, but, personally, I think guaranteeing the minimum bandwidth is more important than limitation of maximum bandwidth to satisfy the requirement in real business sites.

And I know Andrea's io-throttle patch supports the latter case well and it is very stable. But, the first case (guarantee the minimum bandwidth) is not supported in any patches.

Is there any plans to support it? and Is there any problems in implementing it?

I think if IO controller can support guaranteeing the minimum bandwidth and work-conserving mode simultaneously, it more easily satisfies the requirement of the business sites.

Additionally, I didn't understand Proportional bandwidth automatically allows to guarantee minimum requirements and soft constraints.

Can you give me a advice about this? Thanks in advance.

Dong-Jae Kang

I think this is what dm-ioband does.

Let's say you make two groups share the same disk, and give them 70% of the bandwidth the disk physically has and 30% respectively. This means the former group is almost guaranteed to be able to use 70% of the bandwidth even when the latter one is issuing quite a lot of I/O requests.

Yes, I know there exist head seek lags with traditional magnetic disks, so it's important to improve the algorithm to reduce this overhead.

And I think it is also possible to add a new scheduling policy to guarantee the minimum bandwidth. It might be cool if some group can use guaranteed bandwidths and the other share the rest on proportional bandwidth policy.

Thanks,

Hirokazu Takahashi.

With IO limiting approach minimum requirements are supposed to be guaranteed if the user configures a generic block device so that the sum of the limits doesn't exceed the total IO bandwidth of that device. But, in principle, there's nothing in "throttling" that guarantees "fairness" among different cgroups doing IO on the same block devices, that means there's nothing to guarantee minimum requirements (and this is the reason because I liked the Satoshi's CFQ-cgroup approach together with io-throttle).

A more complicated issue is how to evaluate the total IO bandwidth of a generic device. We can use some kind of averaging/prediction, but basically it would be inaccurate due to the mechanic of disks (head seeks, but also caching, buffering mechanisms implemented directly into the device, etc.). It's a hard problem. And the same problem exists also for proportional bandwidth as well, in terms of IO rate predictability I mean.

The only difference is that with proportional bandwidth you know that (taking the same example reported by Hirokazu) with i.e. 10 similar IO requests, 7 will be dispatched to the first cgroup and 3 to the other cgroup. So, you don't need anything to guarantee "fairness", but it's hard also for this case to evaluate the cost of the 7 IO requests respect to the cost of the other 3 IO requests as seen by user applications, that is the cost the users care about.

—Andrea

—

To unsubscribe from this list: send the line "unsubscribe linux-kernel" in the body of a message to majordomo@xxxxxxxxxxxxxxxxx

More majordomo info at <http://vger.kernel.org/majordomo-info.html>

Please read the FAQ at <http://www.tux.org/lkml/>