

---

Subject: Re: [PATCH 0/5] blk-throttle: writeback and swap IO control  
Posted by [KAMEZAWA Hiroyuki](#) on Thu, 24 Feb 2011 00:40:39 GMT  
[View Forum Message](#) <> [Reply to Message](#)

---

On Wed, 23 Feb 2011 19:10:33 -0500  
Vivek Goyal <[vgoyal@redhat.com](mailto:vgoyal@redhat.com)> wrote:

> On Thu, Feb 24, 2011 at 12:14:11AM +0100, Andrea Righi wrote:  
> > On Wed, Feb 23, 2011 at 10:23:54AM -0500, Vivek Goyal wrote:  
> > > > Agreed. Granularity of per inode level might be acceptable in many  
> > > > cases. Again, I am worried faster group getting stuck behind slower  
> > > > group.  
> > > >  
> > > > I am wondering if we are trying to solve the problem of ASYNC write throttling  
> > > > at wrong layer. Should ASYNC IO be throttled before we allow task to write to  
> > > > page cache. The way we throttle the process based on dirty ratio, can we  
> > > > just check for throttle limits also there or something like that. (I think  
> > > > that's what you had done in your initial throttling controller implementation?)  
> > > >  
> > > > Right. This is exactly the same approach I've used in my old throttling  
> > > > controller: throttle sync READs and WRITES at the block layer and async  
> > > > WRITES when the task is dirtying memory pages.  
> > > >  
> > > > This is probably the simplest way to resolve the problem of faster group  
> > > > getting blocked by slower group, but the controller will be a little bit  
> > > > more leaky, because the writeback IO will be never throttled and we'll  
> > > > see some limited IO spikes during the writeback.  
> > > >  
> > > > Yes writeback will not be throttled. Not sure how big a problem that is.  
> > > >  
> > > > - We have controlled the input rate. So that should help a bit.  
> > > > - May be one can put some high limit on root cgroup to in blkio throttle  
> > > > controller to limit overall WRITE rate of the system.  
> > > > - For SATA disks, try to use CFQ which can try to minimize the impact of  
> > > > WRITE.  
> > > >  
> > > > It will atleast provide consistent bandwidth experience to application.  
> >  
> > Right.  
> >  
> > >  
> > > > However, this is always  
> > > > a better solution IMHO respect to the current implementation that is  
> > > > affected by that kind of priority inversion problem.  
> > > >  
> > > > I can try to add this logic to the current blk-throttle controller if  
> > > > you think it is worth to test it.  
> > >

> > > At this point of time I have few concerns with this approach.  
> > >  
> > > - Configuration issues. Asking user to plan for SYNC and ASYNC IO  
> > > separately is inconvenient. One has to know the nature of workload.  
> > >  
> > > - Most likely we will come up with global limits (atleast to begin with),  
> > > and not per device limit. That can lead to contention on one single  
> > > lock and scalability issues on big systems.  
> > >  
> > > Having said that, this approach should reduce the kernel complexity a lot.  
> > > So if we can do some intelligent locking to limit the overhead then it  
> > > will boil down to reduced complexity in kernel vs ease of use to user. I  
> > > guess at this point of time I am inclined towards keeping it simple in  
> > > kernel.  
> > >  
> > >  
> > BTW, with this approach probably we can even get rid of the page  
> > tracking stuff for now.  
>  
> Agreed.  
>  
> > If we don't consider the swap IO, any other IO  
> > operation from our point of view will happen directly from process  
> > context (writes in memory + sync reads from the block device).  
>  
> Why do we need to account for swap IO? Application never asked for swap  
> IO. It is kernel's decision to move some pages to swap to free up some  
> memory. What's the point in charging those pages to application group  
> and throttle accordingly?  
>

I think swap I/O should be controlled by memcg's dirty\_ratio.  
But, IIRC, NEC guy had a requirement for this...

I think some enterprise customer may want to throttle the whole speed of  
swapout I/O (not swapin)...so, they may be glad if they can limit throttle  
the I/O against a disk partition or all I/O tagged as 'swapio' rather than  
some cgroup name.

But I'm afraid slow swapout may consume much dirty\_ratio and make things  
worse ;)

> >  
> > However, I'm sure we'll need the page tracking also for the blkio  
> > controller soon or later. This is an important information and also the  
> > proportional bandwidth controller can take advantage of it.

>  
> Yes page tracking will be needed for CFQ proportional bandwidth ASYNC  
> write support. But until and unless we implement memory cgroup dirty  
> ratio and figure a way out to make writeback logic cgroup aware, till  
> then I think page tracking stuff is not really useful.  
>

I think Greg Thelen is now preparing patches for dirty\_ratio.

Thanks,  
-Kame

---

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

---