Subject: Re: [PATCH 3/5] page_cgroup: make page tracking available for blkio
Posted by Vivek Goyal on Tue, 22 Feb 2011 21:57:20 GMT
View Forum Message <> Reply to Message

On Tue, Feb 22, 2011 at 01:01:45PM -0700, Jonathan Corbet wrote:
> On Tue, 22 Feb 2011 18:12:54 +0100
> Andrea Righi <arighi@develer.com> wrote:
>
> > The page_cgroup infrastructure, currently available only for the memory
> > cgroup controller, can be used to store the owner of each page and
> > opportunely track the writeback IO. This information is encoded in
> > the upper 16-bits of the page_cgroup->flags.
> >
> > A owner can be identified using a generic ID number and the following
> > interfaces are provided to store a retrieve this information:
> >
> > unsigned long page_cgroup_get_owner(struct page *page);
> > int page_cgroup_set_owner(struct page *page, unsigned long id);
> > int page_cgroup_copy_owner(struct page *npage, struct page *opage);
>
> My immediate observation is that you're not really tracking the "owner"
> here - you're tracking an opaque 16-bit token known only to the block
> controller in a field which - if changed by anybody other than the block
> controller - will lead to mayhem in the block controller. I think it
> might be clearer - and safer - to say "blkcg" or some such instead of
> "owner" here.
>
> I'm tempted to say it might be better to just add a pointer to your
> throtl_grp structure into struct page_cgroup.

throtl_grp might not even be present when page is being dirtied. When this
IO is actually submitted to device, we migth end up creating new
throtl_grp. I guess other concern here would be increasing the size of
page_cgroup structure.

I guess you meant storing a pointer to blkio_cgroup, along the lines of
storing a pointer to mem_cgroup. That also means extra 8 bytes and only
one subsystem can use it at a time. So using upper bits of pc->flags
is probably better.

> Or maybe replace the
> mem_cgroup pointer with a single pointer to struct css_set. Both of
> those ideas, though, probably just add unwanted extra overhead now to gain
> generality which may or may not be wanted in the future.

This sounds interesting. IIUC, then this single pointer will allow all
the subsystems to use this single pointer to retireve respective cgroups
without actually co-mounting them.

I am not sure how much work is involved in making it happen. Also not sure about the overhead involved in traversing one extra pointer. Also apart from blkio controller, have we practically felt the need of any other controller this info. (network controller?). Few days back we were experimenting with trying to control block IO bandwidth over NFS with the help of network controller but it did not really work well with host of issues and one them being losing the context information.

If storing css_set pointer is lot of work, may be for the time being we can go for this hardcoding that these bits are exclusively used by blkio controller and once some other controller wants to share it, then look for ways of how to do sharing.

Thanks
Vivek

_____

Containers mailing list
Containers@lists.linux-foundation.org
https://lists.linux-foundation.org/mailman/listinfo/containe rs