Subject: Re: [PATCH -mm 5/5] swapcgroup (v3): implement force_empty
Posted by KAMEZAWA Hiroyuki on Sat, 05 Jul 2008 04:27:31 GMT

View Forum Message <> Reply to Message

On Fri, 4 Jul 2008 21:33:01 +0900
Daisuke Nishimura <nishimura@mxp.nes.nec.co.jp> wrote:

> On Fri, 4 Jul 2008 19:16:38 +0900, KAMEZAWA Hiroyuki <kamezawa.hiroyu@jp.fujitsu.com> wrote:
> > On Fri, 4 Jul 2008 15:24:23 +0900
> > Daisuke Nishimura <nishimura@mxp.nes.nec.co.jp> wrote:
> >
> > > This patch implements force_empty of swapcgroup.
> > >
> > > Currently, it simply uncharges all the charges from the group.
> > >
> > > I think there can be other implementations.
> > >
> > > What I thought are:
> > > - move all the charges to its parent.
> > > - unuse(swap in) all the swap charged to the group.
> > >
> > 3. move all swap back to memory (see swapoff.)
> >
> >
> Do you mean swapping in all the swap including used by
> other groups?

swapping in all swap used by the group (not by all group)

> It would be one choice anyway.
>
> > > But in any case, I think before implementing this way,
> > > hierarchy and move charges support are needed.
> > >
> > > So I think this is enough for the first step.
> > >
> >
> > I don't think hierarchy/automatic-load-balancer for swap cg is necessary.
> It's the problem of how the "hierarchy" would be, I think.

yes.

> I'm saying "hierarchy" here just to mean "some kind of feature
> where a parent includes their children".
> I think "hierarchy" is needed if we implement the choice 1 above,
> and I personally think it would be the best choice.

>

> > Hmm...but handling limit_change (at least, returns -EBUSY) will be necessary.
> I think so too.
> But I'm not sure now it's good or bad to support shrinking at limit_change
> about swap.
> Shrinking swap means increasing the memory usage and that may cause
> another swapout.

yes. but who reduce the limit ? it's the admin or users.

At leaset, returning -EBUSY is necesary. You can use
res_counter: check limit change patch which I posted yesterday.

>
> > Do you consider a some magical way to move pages in swap back to memory ?
> >
> In this patch, I modified the find_next_to_unuse() to find
> the entry charged to a specific group.
> It might be possible to modify try_to_unuse()(or define another function
> based on try_to_unuse()) to reduce swap usage of a specified group
> down to some threashold.
> But, I think, one problem here is from which device the swaps
> should be back to memory, or usage balance between swap devices.
>

Ah, that's maybe difficult one.
As memcg has its own LRU, add MRU to swap ....is not a choice ;(

> > In general, I like this set but we can't change the limit on demand. (maybe)
> > (just putting it to TO-DO-List is okay to me.)
> >
> I'm sorry but what do you mean by "change the limit on demand"?
> Could you explain more?
>

In short, the administrator have to write the perfect plan to set
each group's swap limit beforehand because we cannot decrease used swap.

1st problem is that the user cannot reduce the usage of swap by hand.
(He can reduce by killing process or deleting shmem.)
Once the usage of swap of a group grows, other groups can't use much.

2nd problem is there is no entity who controls the total amount of swap.
The user/admin have to check the amount of free swap space by himself at planning
each group's swap limit more carefully than memcg.

So, I think rich-control of hierarchy will be of no use ;)
All things should be planned before the system starts.

In memcg, the amount of free memory is maintained by global LRU. It does much jobs for us. But free swap space isn't. It's just used on demand.

If we can't decrease usage of swap by a group by hand, the problem which this swap-space-controller want to fix will not be fixed at pleasant level.

Anyway, please return -EBUSY at setting limit < usage, at first :)
That's enough for me, now.

Thanks,
-Kame

____