Subject: Re: [RFC/PATCH] cgroup swap subsystem
Posted by Balbir Singh on Wed, 05 Mar 2008 07:28:54 GMT
View Forum Message <> Reply to Message

Daisuke Nishimura wrote:
> Hi.
>
> Even if limiting memory usage by cgroup memory subsystem
> or isolating memory by cpuset, swap space is shared, so
> resource isolation is not enough. If one group uses up all the
> swap space, it can affect other groups.
>

Yes, that is true. Please ensure that you also cc Hugh Dickins for all swap
related changes.

> I try making a patch of swap subsystem based on memory
> subsystem, which limits swap usage per cgroup.
> It can now charge and limit the swap usage.
>
> I implemented this feature as a new subsystem,
> not as a part of memory subsystem, because I don't want to
> make big change to memcontrol.c, and even if implemented
> as other subsystem, users can manage memory and swap on
> the same cgroup directory if mount them together.
>

I agree, the swap system should be independent of the memory resource controller.

> Basic idea of my implementation:
> - what will be charged ?
> the number of swap entries.
>
> - when to charge/uncharge ?
> charge at get_swap_entry(), and uncharge at swap_entry_free().
>

You mean get_swap_page(), I suppose. The assumption in the code is that every
swap page being charged has already been charged by the memory controller (that
will go against making the controllers independent). Also, be careful of any
charge operations under a spin_lock(). We tried controlling pages in the swap
cache, but Hugh found problems with it, specially due to accounting for pages
that are read ahead to the correct cgroup.

> - to what group charge the swap entry ?
> To determine to what swap_cgroup (corresponding to mem_cgroup in
> memory subsystem) the swap entry should be charged,
> I added a pointer to mm_struct to page_cgroup(pc->pc_mm), and

> changed the argument of get_swap_entry() from (void) to
> (struct page *). As a result, get_swap_entry() can determine
> to what swap_cgroup it should charge the swap entry
> by referring to page->page_cgroup->mm_struct->swap_cgroup.
>

I presume this is for the case when the memory and swap controllers are mounted
in different hierarchies. It seems like too many dereferences to get to the
swap_cgroup

> - from what group uncharge the swap entry ?
> I added to swap_info_struct a member 'struct swap_cgroup **',
> array of pointer to which swap_cgroup the swap entry is
> charged.
>
> Todo:
> - rebase new kernel, and split into some patches.
> - Merge with memory subsystem (if it would be better), or
> remove dependency on CONFIG_CGROUP_MEM_CONT if possible
> (needs to make page_cgroup more generic one).
> - More tests, cleanups, and feartures :-)
>
>
> Any comments or discussions would be appreciated.
>

To be honest, I tried looking at the code, but there were too many #ifdefs and I
sort of lost myself in them.

> Thanks,
> Daisuke Nishimura
>

--
Warm Regards,
Balbir Singh
Linux Technology Center
IBM, ISTL

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

---

Subject: Re: [RFC/PATCH] cgroup swap subsystem
Posted by Daisuke Nishimura on Fri, 07 Mar 2008 04:23:02 GMT
View Forum Message <> Reply to Message

Hi.

Balbir Singh wrote:
> Daisuke Nishimura wrote:
>> Basic idea of my implementation:
>> - what will be charged ?
>>   the number of swap entries.
>>
>> - when to charge/uncharge ?
>>   charge at get_swap_entry(), and uncharge at swap_entry_free().
>>
>
> You mean get_swap_page(), I suppose. The assumption in the code is that every
> swap page being charged has already been charged by the memory controller (that
> will go against making the controllers independent). Also, be careful of any

To make swap-limit independent of memory subsystem, I think
page_cgroup code should be separated into two part:
subsystem-independent and subsystem-dependent, that is
part of associating page and page_cgroup and that of associating
page_cgroup and subsystem.

Rather than to do such a thing, I now think that
it would be better to implement swap-limit as part of
memory subsystem.

Thanks,
Daisuke Nishimura.