

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 features :-)  
>  
>  
> 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.

---

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

---