Subject: Re: [PATCH 0/7] memcg kernel memory tracking
Posted by Ying Han on Thu, 23 Feb 2012 18:18:22 GMT
View Forum Message <> Reply to Message

On Tue, Feb 21, 2012 at 3:34 AM, Glauber Costa <glommer@parallels.com> wrote:
> This is a first structured approach to tracking general kernel
> memory within the memory controller. Please tell me what you think.
>
> As previously proposed, one has the option of keeping kernel memory
> accounted separatedly, or together with the normal userspace memory.
> However, this time I made the option to, in this later case, bill
> the memory directly to memcg->res. It has the disadvantage that it becomes
> complicated to know which memory came from user or kernel, but OTOH,
> it does not create any overhead of drawing from multiple res_counters
> at read time. (and if you want them to be joined, you probably don't care)

Keeping one counter for user and kernel pages makes it easier for
admins to configure the system. About reporting, we should still
report the user and kernel memory separately. It will be extremely
useful when diagnosing the system like heavily memory pressure or OOM.

> Kernel memory is never tracked for the root memory cgroup. This means
> that a system where no memory cgroups exists other than the root, the
> time cost of this implementation is a couple of branches in the slub
> code - none of them in fast paths. At the moment, this works only
> with the slub.
>
> At cgroup destruction, memory is billed to the parent. With no hierarchy,
> this would mean the root memcg. But since we are not billing to that,
> it simply ceases to be tracked.
>
> The caches that we want to be tracked need to explicit register into
> the infrastructure.

It would be hard to let users to register which slab to track
explicitly. We should track them all in general, even with the ones
without shrinker, we want to understand how much is used by which
cgroup.

--Ying

>
> If you would like to give it a try, you'll need one of Frederic's patches
> that is used as a basis for this
> (cgroups: ability to stop res charge propagation on bounded ancestor)
>
> Glauber Costa (7):
> small cleanup for memcontrol.c

```
>  Basic kernel memory functionality for the Memory Controller
>  per-cgroup slab caches
>  chained slab caches: move pages to a different cache when a cache is
>    destroyed.
>  shrink support for memcg kmem controller
>  track dcache per-memcg
>  example shrinker for memcg-aware dcache
>
>  fs/dcache.c                |  136 +++++++++++++++++-
>  include/linux/dcache.h     |    4 +
>  include/linux/memcontrol.h |   35 +++++
>  include/linux/shrinker.h   |    4 +
>  include/linux/slab.h       |   12 ++
>  include/linux/slub_def.h   |    3 +
>  mm/memcontrol.c            |  344
+++++++++++++++++++++++++++++++++++++++++++++-
>  mm/slub.c                  |  237 +++++++++++++++++++++++++++++---
>  mm/vmscan.c                |   60 +++++++++-
>  9 files changed, 806 insertions(+), 29 deletions(-)
>
> --
> 1.7.7.6
>
> --
> To unsubscribe, send a message with 'unsubscribe linux-mm' in
> the body to majordomo@kvack.org.  For more info on Linux MM,
> see: http://www.linux-mm.org/ .
> Fight unfair telecom internet charges in Canada: sign http://stopthemeter.ca/
> Don't email: <a href=mailto:"dont@kvack.org"> email@kvack.org </a>
```