Subject: Re: [PATCH v6 1/8] Basic kernel memory functionality for the Memory Controller
Posted by Greg Thelen on Thu, 13 Oct 2011 07:18:49 GMT
View Forum Message <> Reply to Message

On Mon, Oct 10, 2011 at 3:24 AM, Glauber Costa <glommer@parallels.com> wrote:
> diff --git a/Documentation/cgroups/memory.txt b/Documentation/cgroups/memory.txt
> index 06eb6d9..bf00cd2 100644
> --- a/Documentation/cgroups/memory.txt
> +++ b/Documentation/cgroups/memory.txt
...
> @@ -255,6 +262,31 @@ When oom event notifier is registered, event will be delivered.
> per-zone-per-cgroup LRU (cgroup's private LRU) is just guarded by
> zone->lru_lock, it has no lock of its own.
>
> +2.7 Kernel Memory Extension (CONFIG_CGROUP_MEM_RES_CTLR_KMEM)
> +
> + With the Kernel memory extension, the Memory Controller is able to limit

Extra leading space before 'With'.

> +the amount of kernel memory used by the system. Kernel memory is fundamentally
> +different than user memory, since it can't be swapped out, which makes it
> +possible to DoS the system by consuming too much of this precious resource.
> +Kernel memory limits are not imposed for the root cgroup.
> +
> +Memory limits as specified by the standard Memory Controller may or may not
> +take kernel memory into consideration. This is achieved through the file
> +memory.independent_kmem_limit. A Value different than 0 will allow for kernel

s/Value/value/

> diff --git a/mm/memcontrol.c b/mm/memcontrol.c
> index 3508777..d25c5cb 100644
> --- a/mm/memcontrol.c
> +++ b/mm/memcontrol.c
...
> +static int kmem_limit_independent_write(struct cgroup *cont, struct cftype *cft,
> +   u64 val)
> +{
> +   cgroup_lock();
> +   mem_cgroup_from_cont(cont)->kmem_independent_accounting = !!val;
> +   cgroup_unlock();

I do not think cgroup_lock,unlock are needed here. The cont and
associated cgroup should be guaranteed by the caller to be valid.
Does this lock provide some other synchronization?