

---

Subject: Re: [RFC][for -mm] memory controller enhancements for reclaiming take2  
[7/8] bacground reclaim for m

Posted by [yamamoto](#) on Tue, 04 Dec 2007 03:07:55 GMT

[View Forum Message](#) <> [Reply to Message](#)

---

```
> @@ -1186,6 +1251,16 @@ static void free_mem_cgroup_per_zone_inf
>
> static struct mem_cgroup init_mem_cgroup;
>
> +static int __init mem_cgroup_reclaim_init(void)
> +{
> + init_mem_cgroup.daemon.thread = kthread_run(mem_cgroup_reclaim_daemon,
> +     &init_mem_cgroup, "memcontd");
> + if (IS_ERR(init_mem_cgroup.daemon.thread))
> + BUG();
> + return 0;
> +}
> +late_initcall(mem_cgroup_reclaim_init);
> +
> static struct cgroup_subsys_state *
> mem_cgroup_create(struct cgroup_subsys *ss, struct cgroup *cont)
> {
> @@ -1213,6 +1288,17 @@ mem_cgroup_create(struct cgroup_subsys *
>     if (alloc_mem_cgroup_per_zone_info(mem, node))
>         goto free_out;
>
> + /* Memory Reclaim Daemon per cgroup */
> + init_waitqueue_head(&mem->daemon.waitq);
> + if (mem != &init_mem_cgroup) {
> + /* Complicated...but we cannot call kthread create here..*/
> + /* init call will later assign kthread */
> + mem->daemon.thread = kthread_run(mem_cgroup_reclaim_daemon,
> +     mem, "memcontd");
> + if (IS_ERR(mem->daemon.thread))
> +     goto free_out;
> +}
> +
> return &mem->css;
> free_out:
> for_each_node_state(node, N_POSSIBLE)
```

you don't need the kthread as far as RES\_HWMARK is "infinite".  
given the current default value of RES\_HWMARK, you can simplify  
initialization by deferring the kthread creation to mem\_cgroup\_write.

YAMAMOTO Takashi

---

Containers mailing list

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

---