Subject: Re: [PATCH v2 02/11] memcg: Reclaim when more than one page needed.
Posted by Michal Hocko on Fri, 10 Aug 2012 15:42:40 GMT
View Forum Message <> Reply to Message

On Thu 09-08-12 17:01:10, Glauber Costa wrote:
[...]
> @@ -2317,18 +2318,18 @@ static int mem_cgroup_do_charge(struct mem_cgroup *memcg, gfp_t gfp_mask,
> } else
> mem_over_limit = mem_cgroup_from_res_counter(fail_res, res);
> /*
> - * nr_pages can be either a huge page (HPAGE_PMD_NR), a batch
> - * of regular pages (CHARGE_BATCH), or a single regular page (1).
> - *
> * Never reclaim on behalf of optional batching, retry with a
> * single page instead.
> */
> - if (nr_pages == CHARGE_BATCH)
> + if (nr_pages > min_pages)
> return CHARGE_RETRY;

This is dangerous because THP charges will be retried now while they previously failed with CHARGE_NOMEM which means that we will keep attempting potentially endlessly.
Why cannot we simply do if (nr_pages < CHARGE_BATCH) and get rid of the min_pages altogether?
Also the comment doesn't seem to be valid anymore.

>
> if (!(gfp_mask & __GFP_WAIT))
> return CHARGE_WOULDBLOCK;
>
> + if (gfp_mask & __GFP_NORETRY)
> + return CHARGE_NOMEM;
> +
> ret = mem_cgroup_reclaim(mem_over_limit, gfp_mask, flags);
> if (mem_cgroup_margin(mem_over_limit) >= nr_pages)
> return CHARGE_RETRY;
> @@ -2341,7 +2342,7 @@ static int mem_cgroup_do_charge(struct mem_cgroup *memcg, gfp_t gfp_mask,
> * unlikely to succeed so close to the limit, and we fall back
> * to regular pages anyway in case of failure.
> */
> - if (nr_pages == 1 && ret)
> + if (nr_pages <= (1 << PAGE_ALLOC_COSTLY_ORDER) && ret)
> return CHARGE_RETRY;
>

```
>   /*
> @@ -2476,7 +2477,8 @@ again:
>     nr_oom_retries = MEM_CGROUP_RECLAIM_RETRIES;
>    }
>
> -  ret = mem_cgroup_do_charge(memcg, gfp_mask, batch, oom_check);
> +  ret = mem_cgroup_do_charge(memcg, gfp_mask, batch, nr_pages,
> +      oom_check);
>    switch (ret) {
>    case CHARGE_OK:
>     break;
> --
> 1.7.11.2
>
> --
> To unsubscribe from this list: send the line "unsubscribe cgroups" in
> the body of a message to majordomo@vger.kernel.org
> More majordomo info at  http://vger.kernel.org/majordomo-info.html

--
Michal Hocko
SUSE Labs
```