Subject: Re: [PATCH v4 06/25] memcg: Make it possible to use the stock for more than one page.
Posted by Michal Hocko on Thu, 21 Jun 2012 21:14:52 GMT
View Forum Message <> Reply to Message

```
On Wed 20-06-12 23:36:47, Glauber Costa wrote:
> On 06/20/2012 05:28 PM, Michal Hocko wrote:
> >On Mon 18-06-12 14:27:59, Glauber Costa wrote:
> >>From: Suleiman Souhlal <ssouhlal@FreeBSD.org>
> >>
> >>Signed-off-by: Suleiman Souhlal <suleiman@google.com>
> >>Signed-off-by: Glauber Costa <glommer@parallels.com>
> >>Acked-by: Kamezawa Hiroyuki <kamezawa.hiroyu@jp.fujitsu.com>
> >
> >I am not sure the patch is good to merge on its own without the rest.
> >One comment bellow.
> >
> >>---
> >>  mm/memcontrol.c |   18 +++++++++---------
> >>  1 file changed, 9 insertions(+), 9 deletions(-)
> >>
> >>diff --git a/mm/memcontrol.c b/mm/memcontrol.c
> >>index ce15be4..00b9f1e 100644
> >>--- a/mm/memcontrol.c
> >>+++ b/mm/memcontrol.c
> >>@@ -1998,19 +1998,19 @@ static DEFINE_PER_CPU(struct memcg_stock_pcp,
memcg_stock);
> >>  static DEFINE_MUTEX(percpu_charge_mutex);
> >>
> >>  /*
> >>- * Try to consume stocked charge on this cpu. If success, one page is consumed
> >>- * from local stock and true is returned. If the stock is 0 or charges from a
> >>- * cgroup which is not current target, returns false. This stock will be
> >>- * refilled.
> >>+ * Try to consume stocked charge on this cpu. If success, nr_pages pages are
> >>+ * consumed from local stock and true is returned. If the stock is 0 or
> >>+ * charges from a cgroup which is not current target, returns false.
> >>+ * This stock will be refilled.
> >>   */
> >>-static bool consume_stock(struct mem_cgroup *memcg)
> >>+static bool consume_stock(struct mem_cgroup *memcg, int nr_pages)
> >>  {
> >>   struct memcg_stock_pcp *stock;
> >>   bool ret = true;
> >
> >I guess you want:
> > if (nr_pages > CHARGE_BATCH)
> >  return false;
```

> >
> >because you don't want to try to use stock for THP pages.
>
>
> The code reads:
>
> +   if (memcg == stock->cached && stock->nr_pages >= nr_pages)
> +   stock->nr_pages -= nr_pages;
>
> Isn't stock->nr_pages always <= CHARGE_BATCH by definition?

Yes it is, but why to disable preemption if we know this has no chance
to succeed at all?

--
Michal Hocko
SUSE Labs
SUSE LINUX s.r.o.
Lihovarska 1060/12
190 00 Praha 9
Czech Republic