Subject: Re: [RFC][PATCH] memory cgroup enhancements updated [10/10] NUMA aware account
Posted by KAMEZAWA Hiroyuki on Wed, 24 Oct 2007 15:53:04 GMT
View Forum Message <> Reply to Message

On Wed, 24 Oct 2007 20:29:08 +0530
Balbir Singh <balbir@linux.vnet.ibm.com> wrote:

> > + for_each_possible_cpu(cpu) {
> > + int nid = cpu_to_node(cpu);
> > + struct mem_cgroup_stat_cpu *mcsc;
> > + if (sizeof(*mcsc) < PAGE_SIZE)
> > +   mcsc = kmalloc_node(sizeof(*mcsc), GFP_KERNEL, nid);
> > + else
> > +   mcsc = vmalloc_node(sizeof(*mcsc), nid);
>
> Do we need to use the vmalloc() pool? I think we might be better off
> using a dedicated slab for ourselves
>
I admit this part is complicated. But ia64's MAX_NUMNODES=1024 and stat
can be increased. we need vmalloc. I'll rewrite this part to be
better looking.

> > + memset(mcsc, 0, sizeof(*mcsc));
> > + mem->stat.cpustat[cpu] = mcsc;
> > + }
> >   return &mem->css;
> > }
> >
> > @@ -969,7 +1006,15 @@ static void mem_cgroup_pre_destroy(struc
> > static void mem_cgroup_destroy(struct cgroup_subsys *ss,
> >   struct cgroup *cont)
> > {
> > - kfree(mem_cgroup_from_cont(cont));
> > + struct mem_cgroup *mem = mem_cgroup_from_cont(cont);
> > + int cpu;
> > + for_each_possible_cpu(cpu) {
> > + if (sizeof(struct mem_cgroup_stat_cpu) < PAGE_SIZE)
> > +   kfree(mem->stat.cpustat[cpu]);
> > + else
> > +   vfree(mem->stat.cpustat[cpu]);
> > + }
> > + kfree(mem);
> > }
> >
> > static int mem_cgroup_populate(struct cgroup_subsys *ss,
> > @@ -1021,5 +1066,5 @@ struct cgroup_subsys mem_cgroup_subsys =
> >   .destroy = mem_cgroup_destroy,

> > .populate = mem_cgroup_populate,
> > .attach = mem_cgroup_move_task,
> > - .early_init = 1,
> > + .early_init = 0,
>
> I don't understand why this change is required here?
>
If early_init = 1, we cannot call kmalloc/vmalloc at initializing init_mem_cgroup.
It's too early.

Thanks,
-Kame

_____
Containers mailing list
Containers@lists.linux-foundation.org
https://lists.linux-foundation.org/mailman/listinfo/containers