
Subject: Re: [PATCH v7 02/10] foundations of per-cgroup memory pressure controlling.

Posted by [Glauber Costa](#) on Fri, 02 Dec 2011 17:46:46 GMT

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

```
>> static void proto_seq_printf(struct seq_file *seq, struct proto *proto)
>> {
>> + struct mem_cgroup *memcg = mem_cgroup_from_task(current);
>> +
>> seq_printf(seq, "%-9s %4u %6d %6ld %-3s %6u %-3s %-10s "
>> "%2c %2c %2c %2c %2c %2c %2c %2c %2c %2c %2c %2c %2c %2c %2c %2c %2c %2c %2c %2c %2c\n",
>> proto->name,
>> proto->obj_size,
>> sock_prot_inuse_get(seq_file_net(seq), proto),
>> - proto->memory_allocated != NULL ? atomic_long_read(proto->memory_allocated) : -1L,
>> - proto->memory_pressure != NULL ? *proto->memory_pressure ? "yes" : "no" : "NI",
>> + sock_prot_memory_allocated(proto, memcg),
>> + sock_prot_memory_pressure(proto, memcg),
>
> I wonder I should say NO, here. (Networking guys are ok ??)
>
> IIUC, this means there is no way to see aggregated sockstat of all system.
> And the result depends on the cgroup which the caller is under control.
>
> I think you should show aggregated sockstat(global + per-memcg) here and
> show per-memcg ones via /cgroup interface or add private_sockstat to show
> per cgroup summary.
>
```

Hi Kame,

Yes, the statistics displayed depends on which cgroup you live. Also, note that the parent cgroup here is always updated (even when use_hierarchy is set to 0). So it is always possible to grab global statistics, by being in the root cgroup.

For the others, I believe it to be a question of naturalization. Any tool that is fetching these values is likely interested in the amount of resources available/used. When you are on a cgroup, the amount of resources available/used changes, so that's what you should see.

Also brings the point of resource isolation: if you shouldn't interfere with other set of process' resources, there is no reason for you to see them in the first place.

So given all that, I believe that whenever we talk about resources in a cgroup, we should talk about cgroup-local ones.
