Subject: Re: [PATCH v7 02/10] foundations of per-cgroup memory pressure controlling.
Posted by KAMEZAWA Hiroyuki on Wed, 30 Nov 2011 00:43:05 GMT
View Forum Message <> Reply to Message

On Tue, 29 Nov 2011 21:56:53 -0200
Glauber Costa <glommer@parallels.com> wrote:

```
> This patch replaces all uses of struct sock fields' memory_pressure,
> memory_allocated, sockets_allocated, and sysctl_mem to acessor
> macros. Those macros can either receive a socket argument, or a mem_cgroup
> argument, depending on the context they live in.
>
> Since we're only doing a macro wrapping here, no performance impact at all is
> expected in the case where we don't have cgroups disabled.
>
> Signed-off-by: Glauber Costa <glommer@parallels.com>
> CC: David S. Miller <davem@davemloft.net>
> CC: Hiroyouki Kamezawa <kamezawa.hiroyu@jp.fujitsu.com>
> CC: Eric W. Biederman <ebiederm@xmission.com>
> CC: Eric Dumazet <eric.dumazet@gmail.com>
<snip>

> +static inline bool
> +memcg_memory_pressure(struct proto *prot, struct mem_cgroup *memcg)
> +{
> + if (!prot->memory_pressure)
> +  return false;
> + return !!prot->memory_pressure;
> +}
```

I think you should take a deep breath and write patech relaxedly, and do enough test.

This should be

```
 return !!*prot->memory_pressure;
```

BTW, I don't like to receive tons of everyday-update even if you're in hurry.

```
>  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\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.

Thanks,
-Kame