Subject: Re: [PATCH 2/3] don't take cgroup_mutex in destroy()
Posted by Glauber Costa on Fri, 20 Apr 2012 15:04:11 GMT
View Forum Message <> Reply to Message

On 04/19/2012 07:57 PM, Tejun Heo wrote:
> On Thu, Apr 19, 2012 at 07:49:17PM -0300, Glauber Costa wrote:
>> Most of the destroy functions are only doing very simple things
>> like freeing memory.
>>
>> The ones who goes through lists and such, already use its own
>> locking for those.
>>
>> * The cgroup itself won't go away until we free it, (after destroy)
>> * The parent won't go away because we hold a reference count
>> * There are no more tasks in the cgroup, and the cgroup is declared
>>   dead (cgroup_is_removed() == true)
>>
>> For the blk-cgroup and the cpusets, I got the impression that the mutex
>> is still necessary.
>>
>> For those, I grabbed it from within the destroy function itself.
>>
>> If the maintainer for those subsystems consider it safe to remove
>> it, we can discuss it separately.
>
> I really don't like cgroup_lock() usage spreading more. It's
> something which should be contained in cgroup.c proper. I looked at
> the existing users a while ago and they seemed to be compensating
> deficencies in API, so, if at all possible, let's not spread the
> disease.

Well, I can dig deeper and see if they are really needed. I don't know
cpusets and blkcg *that* well, that's why I took them there, hoping that
someone could enlighten me, maybe they aren't really needed even now.

I agree with the compensating: As I mentioned, most of them are already
taking other kinds of lock to protect their structures, which is the
right thing to do.

There were only two or three spots in cpusets and blkcg where I wasn't
that sure that we could drop the lock... What do you say about that ?