Subject: Re: [PATCH] cgroup: support checking of subsystem dependencies (v2)
Posted by Li Zefan on Tue, 01 Jul 2008 08:43:30 GMT
View Forum Message <> Reply to Message

>> +int subsys_depend(struct cgroup_subsys *ss, unsigned long subsys_bits)
>> +
>> +Called when a cgroup subsystem wants to check if some other subsystems
>> +are also present in the proposed hierarchy. If this method returns error,
>> +the mount of the cgroup filesystem will fail.
>
> OK, but the name subsys_depend is quite poor.
>
> check_subsys_dependency is better. But it still has the failing that
> the reader cannot determine the sense of the function's return value
> from its name. Does it return true on success, or false?
>

Other callbacks return -errno on failure and 0 on success, so I make this new
callback behave the same, and the errno will be returned by the mount command.

But it seems true/false is more reasonable, then if it returns false, mount can
return EINVAL.

I'll document the return value.

> A good name would be something like subsys_dependencies_ok(). Then
> code such as
>
> if (subsys_dependencies_ok(...))
> go_wild();
> else
> bad_hair_day();
>
> makes more sense.
>

Seems better.

>
>> 4. Questions
>> ============
>>
>> diff --git a/include/linux/cgroup.h b/include/linux/cgroup.h
>> index e155aa7..fc99ba4 100644
>> --- a/include/linux/cgroup.h
>> +++ b/include/linux/cgroup.h
>> @@ -305,6 +305,8 @@ struct cgroup_subsys {
>> struct cgroup *cgrp);

```
>>   void (*post_clone)(struct cgroup_subsys *ss, struct cgroup *cgrp);
>>   void (*bind)(struct cgroup_subsys *ss, struct cgroup *root);
>> + int (*subsys_depend)(struct cgroup_subsys *ss,
>> +         unsigned long subsys_bits);
>>   /*
>>    * This routine is called with the task_lock of mm->owner held
>>    */
>> diff --git a/kernel/cgroup.c b/kernel/cgroup.c
>> index 15ac0e1..18e8132 100644
>> --- a/kernel/cgroup.c
>> +++ b/kernel/cgroup.c
>> @@ -774,6 +774,25 @@ static int cgroup_show_options(struct seq_file *seq, struct vfsmount
*vfs)
>>   return 0;
>>  }
>>
>> +static int check_subsys_dependency(unsigned long subsys_bits)
>
> Would be nice to have a little comment explaining this function's role
> in the world.  It should document the meaning of the return values.
>
> Perhaps it could return bool.  That depends upon a well-chosen name,
> and upon the thus-far-undocumented return-value meaning.
>

will fix.

>> +{
>> + int i;
>> + int ret;
>> + struct cgroup_subsys *ss;
>> +
>> + for (i = 0; i < CGROUP_SUBSYS_COUNT; i++) {
>> +  ss = subsys[i];
>> +
>> +  if (test_bit(i, &subsys_bits) && ss->subsys_depend) {
>> +   ret = ss->subsys_depend(ss, subsys_bits);
>> +   if (ret)
>> +    return ret;
>> +  }
>> + }
>> +
>> + return 0;
>> +}
>
>
>>  struct cgroup_sb_opts {
>>   unsigned long subsys_bits;
```

>>   unsigned long flags;
>> @@ -834,7 +853,7 @@ static int parse_cgroupfs_options(char *data,
>>   if (!opts->subsys_bits)
>>   return -EINVAL;
>>
>> - return 0;
>> + return check_subsys_dependency(opts->subsys_bits);
>> }
>
> The whole patch doesn't do anything. Perhaps there's another patch in
> the pipeline somewhere which adds one or more ->subsys_depend
> implementations, but I cannot find it. If so, I'd have expected this
> patch to be titled [1/N].
>

Yes, the patch does nothing actually. Daisuke Nishimura-san's original swap controller needed to be mounted with memory controller, but the newer version makes it an addon to memory controller, so it currently doesn't need the feature this patch provides.

Or I can keep this patch until some new cgroup subsystem needs it ?