
Subject: [PATCH] cgroup: support checking of subsystem dependencies (v2)

Posted by [Li Zefan](#) on Thu, 19 Jun 2008 01:51:36 GMT

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

This allows one subsystem to require that it only be mounted when some other subsystems are also present in the proposed hierarchy.

For example if subsystem foo depends on bar, the following will fail:

```
# mount -t cgroup -ofoo xxx /dev/cgroup
```

You should mount with both subsystems:

```
# mount -t cgroup -ofoo,bar xxx /dev/cgroup
```

foo may implement the `subsys_depend()` callback this way:

```
static int foo_cgroup_subsys_depend(struct cgroup_subsys *ss,
    unsigned long subsys_bits)
{
    if (!test_bit(bar_cgroup_subsys_id, &subsys_bits))
        return -EINVAL;
    return 0;
}
```

Changelog:

- call `check_subsys_depend()` in `parse_cgroupfs_options()`, but not in mount and remount code.

Signed-off-by: Li Zefan <lizf@cn.fujitsu.com>

```
Documentation/cgroups.txt | 6 ++++++
include/linux/cgroup.h    | 2 ++
kernel/cgroup.c           | 21 ++++++-----
3 files changed, 28 insertions(+), 1 deletions(-)
```

```
diff --git a/Documentation/cgroups.txt b/Documentation/cgroups.txt
```

```
index 824fc02..8252f5b 100644
```

```
--- a/Documentation/cgroups.txt
```

```
+++ b/Documentation/cgroups.txt
```

```
@@ -530,6 +530,12 @@ and root cgroup. Currently this will only involve movement between
the default hierarchy (which never has sub-cgroups) and a hierarchy
that is being created/destroyed (and hence has no sub-cgroups).
```

```
+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.
```

```
+
```

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)
+{
+ 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);
}
```

```
static int cgroup_remount(struct super_block *sb, int *flags, char *data)
```

```
--
```

1.5.4.rc3

Containers mailing list

Containers@lists.linux-foundation.org

<https://lists.linux-foundation.org/mailman/listinfo/containers>
