Subject: [PATCH 1/3] cgroup - removing superfluous rcu_read_lock_held check
Posted by Jiri Olsa on Mon, 01 Nov 2010 19:15:07 GMT
View Forum Message <> Reply to Message

hi,
the rcu_dereference_check is defined as

```
 #define rcu_dereference_check(p, c) \
    __rcu_dereference_check((p), rcu_read_lock_held() || (c), __rcu)
```

so the caller does not need to specify rcu_read_lock_held() condition.

wbr,
jirka

Signed-off-by: Jiri Olsa <jolsa@redhat.com>

```
 include/linux/cgroup.h |    1 -
 kernel/cgroup.c        |    6 ++----
 2 files changed, 2 insertions(+), 5 deletions(-)

diff --git a/include/linux/cgroup.h b/include/linux/cgroup.h
index ed4ba11..caed568 100644
--- a/include/linux/cgroup.h
+++ b/include/linux/cgroup.h
@@ -536,7 +536,6 @@ static inline struct cgroup_subsys_state *cgroup_subsys_state(
  */
 #define task_subsys_state_check(task, subsys_id, __c)   \
  rcu_dereference_check(task->cgroups->subsys[subsys_id],  \
-         rcu_read_lock_held() ||   \
          lockdep_is_held(&task->alloc_lock) || \
          cgroup_lock_is_held() || (__c))

diff --git a/kernel/cgroup.c b/kernel/cgroup.c
index 66a416b..1f329a2 100644
--- a/kernel/cgroup.c
+++ b/kernel/cgroup.c
@@ -1687,7 +1687,6 @@ int cgroup_path(const struct cgroup *cgrp, char *buf, int buflen)
 {
  char *start;
  struct dentry *dentry = rcu_dereference_check(cgrp->dentry,
-           rcu_read_lock_held() ||
            cgroup_lock_is_held());

  if (!dentry || cgrp == dummytop) {
@@ -1713,7 +1712,6 @@ int cgroup_path(const struct cgroup *cgrp, char *buf, int buflen)
    break;
```

```
   dentry = rcu_dereference_check(cgrp->dentry,
-                  rcu_read_lock_held() ||
                   cgroup_lock_is_held());
   if (!cgrp->parent)
    continue;
@@ -4544,7 +4542,7 @@ unsigned short css_id(struct cgroup_subsys_state *css)
   * it's unchanged until freed.
   */
  cssid = rcu_dereference_check(css->id,
-   rcu_read_lock_held() || atomic_read(&css->refcnt));
+   atomic_read(&css->refcnt));

  if (cssid)
   return cssid->id;
@@ -4557,7 +4555,7 @@ unsigned short css_depth(struct cgroup_subsys_state *css)
  struct css_id *cssid;

  cssid = rcu_dereference_check(css->id,
-   rcu_read_lock_held() || atomic_read(&css->refcnt));
+   atomic_read(&css->refcnt));

  if (cssid)
   return cssid->depth;
--
1.7.1
```

_____
Containers mailing list
Containers@lists.linux-foundation.org
https://lists.linux-foundation.org/mailman/listinfo/containe rs