

---

Subject: [PATCH -mm 3/3] cgroup: remove the css\_set linked-list  
Posted by [Li Zefan](#) on Wed, 02 Apr 2008 02:16:58 GMT  
[View Forum Message](#) <> [Reply to Message](#)

---

Now we can run through the hash table instead of running through the linked-list.

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

---  
include/linux/cgroup.h | 6 -----  
kernel/cgroup.c | 40 ++++++-----  
2 files changed, 20 insertions(+), 26 deletions(-)

diff --git a/include/linux/cgroup.h b/include/linux/cgroup.h  
index c15c5e0..72f3911 100644

--- a/include/linux/cgroup.h  
+++ b/include/linux/cgroup.h  
@@ -150,12 +150,6 @@ struct css\_set {  
 struct kref ref;

/\*  
- \* List running through all cgroup groups. Protected by  
- \* css\_set\_lock  
- \*/  
- struct list\_head list;  
-  
- /\*  
- \* List running through all cgroup groups in the same hash  
- \* slot. Protected by css\_set\_lock  
- \*/

diff --git a/kernel/cgroup.c b/kernel/cgroup.c  
index 2b72346..c444e6e 100644

--- a/kernel/cgroup.c  
+++ b/kernel/cgroup.c  
@@ -241,7 +241,6 @@ static void unlink\_css\_set(struct css\_set \*cg)  
{  
 write\_lock(&css\_set\_lock);  
 hlist\_del(&cg->hlist);  
- list\_del(&cg->list);  
 css\_set\_count--;  
 while (!list\_empty(&cg->cg\_links)) {  
 struct cg\_cgroup\_link \*link;  
@@ -476,8 +475,6 @@ static struct css\_set \*find\_css\_set(  
  
 BUG\_ON(!list\_empty(&tmp\_cg\_links));

- /\* Link this cgroup group into the list \*/  
- list\_add(&res->list, &init\_css\_set.list);

```

css_set_count++;

/* Add this cgroup group to the hash table */
@@ -962,7 +959,7 @@ static int cgroup_get_sb(struct file_system_type *fs_type,
    int ret = 0;
    struct super_block *sb;
    struct cgroupfs_root *root;
- struct list_head tmp_cg_links, *l;
+ struct list_head tmp_cg_links;
    INIT_LIST_HEAD(&tmp_cg_links);

/* First find the desired set of subsystems */
@@ -1004,6 +1001,7 @@ static int cgroup_get_sb(struct file_system_type *fs_type,
/* New superblock */
    struct cgroup *cgrp = &root->top_cgroup;
    struct inode *inode;
+ int i;

    BUG_ON(sb->s_root != NULL);

@@ -1048,22 +1046,25 @@ static int cgroup_get_sb(struct file_system_type *fs_type,
/* Link the top cgroup in this hierarchy into all
 * the css_set objects */
    write_lock(&css_set_lock);
- l = &init_css_set.list;
- do {
+ for (i = 0; i < CSS_SET_TABLE_SIZE; i++) {
+ struct hlist_head *hhead = &css_set_table[i];
+ struct hlist_node *node;
    struct css_set *cg;
- struct cg_cgroup_link *link;
- cg = list_entry(l, struct css_set, list);
- BUG_ON(list_empty(&tmp_cg_links));
- link = list_entry(tmp_cg_links.next,
- struct cg_cgroup_link,
- cgrp_link_list);
- list_del(&link->cgrp_link_list);
- link->cg = cg;
- list_add(&link->cgrp_link_list,
- &root->top_cgroup.css_sets);
- list_add(&link->cg_link_list, &cg->cg_links);
- l = l->next;
- } while (l != &init_css_set.list);
+
+ hlist_for_each_entry(cg, node, hhead, hlist) {
+ struct cg_cgroup_link *link;
+
+ BUG_ON(list_empty(&tmp_cg_links));

```

```

+ link = list_entry(tmp_cg_links.next,
+   struct cg_cgroup_link,
+   cgrp_link_list);
+ list_del(&link->cgrp_link_list);
+ link->cg = cg;
+ list_add(&link->cgrp_link_list,
+   &root->top_cgroup.css_sets);
+ list_add(&link->cg_link_list, &cg->cg_links);
+ }
+ }
  write_unlock(&css_set_lock);

  free_cg_links(&tmp_cg_links);
@@ -2499,7 +2500,6 @@ int __init cgroup_init_early(void)
  int i;
  kref_init(&init_css_set.ref);
  kref_get(&init_css_set.ref);
- INIT_LIST_HEAD(&init_css_set.list);
  INIT_LIST_HEAD(&init_css_set.cg_links);
  INIT_LIST_HEAD(&init_css_set.tasks);
  INIT_HLIST_NODE(&init_css_set.hlist);
--
1.5.4.rc3

```

---

Containers mailing list  
Containers@lists.linux-foundation.org  
<https://lists.linux-foundation.org/mailman/listinfo/containers>

---