
Subject: [RFC/PATCH 8/8]: CGroup Files: Convert res_counter_write() to be a cgroups write_string() handler

Posted by [Paul Menage](#) on Tue, 13 May 2008 06:37:15 GMT

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

Currently read_counter_write() is a raw file handler even though it's ultimately taking a number, since in some cases it wants to pre-process the string when converting it to a number.

This patch converts res_counter_write() from a raw file handler to a write_string() handler; this allows some of the boilerplate copying/locking/checking to be removed, and simplifies the cleanup path, since these functions are now performed by the cgroups framework.

Signed-off-by: Paul Menage <menage@google.com>

```
---
include/linux/res_counter.h | 4 +--
kernel/res_counter.c       | 36 ++++++++-----
mm/memcontrol.c           | 11 +++++-----
3 files changed, 16 insertions(+), 35 deletions(-)
```

Index: cgroup-2.6.25-mm1/include/linux/res_counter.h

```
=====
--- cgroup-2.6.25-mm1.orig/include/linux/res_counter.h
+++ cgroup-2.6.25-mm1/include/linux/res_counter.h
@@ -63,8 +63,8 @@ u64 res_counter_read_u64(struct res_coun
 ssize_t res_counter_read(struct res_counter *counter, int member,
     const char __user *buf, size_t nbytes, loff_t *pos,
     int (*read_strategy)(unsigned long long val, char *s));
-ssize_t res_counter_write(struct res_counter *counter, int member,
-    const char __user *buf, size_t nbytes, loff_t *pos,
+int res_counter_write(struct res_counter *counter, int member,
+    char *buffer,
+    int (*write_strategy)(char *buf, unsigned long long *val));
```

/*

Index: cgroup-2.6.25-mm1/kernel/res_counter.c

```
=====
--- cgroup-2.6.25-mm1.orig/kernel/res_counter.c
+++ cgroup-2.6.25-mm1/kernel/res_counter.c
@@ -102,44 +102,26 @@ u64 res_counter_read_u64(struct res_coun
     return *res_counter_member(counter, member);
 }

-ssize_t res_counter_write(struct res_counter *counter, int member,
-    const char __user *userbuf, size_t nbytes, loff_t *pos,
-    int (*write_strategy)(char *st_buf, unsigned long long *val))
```

```

+int res_counter_write(
+ struct res_counter *counter, int member,
+ char *buf,
+ int (*write_strategy)(char *st_buf, unsigned long long *val))
{
- int ret;
- char *buf, *end;
+ char *end;
  unsigned long flags;
  unsigned long long tmp, *val;

- buf = kmalloc(nbytes + 1, GFP_KERNEL);
- ret = -ENOMEM;
- if (buf == NULL)
- goto out;
-
- buf[nbytes] = '\0';
- ret = -EFAULT;
- if (copy_from_user(buf, userbuf, nbytes))
- goto out_free;
-
- ret = -EINVAL;
-
- strstrip(buf);
  if (write_strategy) {
- if (write_strategy(buf, &tmp)) {
- goto out_free;
- }
+ if (write_strategy(buf, &tmp))
+ return -EINVAL;
  } else {
    tmp = simple_strtoull(buf, &end, 10);
    if (*end != '\0')
- goto out_free;
+ return -EINVAL;
  }
  spin_lock_irqsave(&counter->lock, flags);
  val = res_counter_member(counter, member);
  *val = tmp;
  spin_unlock_irqrestore(&counter->lock, flags);
- ret = nbytes;
-out_free:
- kfree(buf);
-out:
- return ret;
+ return 0;
}

```

Index: cgroup-2.6.25-mm1/mm/memcontrol.c

```

=====
--- cgroup-2.6.25-mm1.orig/mm/memcontrol.c
+++ cgroup-2.6.25-mm1/mm/memcontrol.c
@@ -848,13 +848,12 @@ static u64 mem_cgroup_read(struct cgroup
    cft->private);
}

-static ssize_t mem_cgroup_write(struct cgroup *cont, struct cftype *cft,
-    struct file *file, const char __user *userbuf,
-    size_t nbytes, loff_t *ppos)
+static int mem_cgroup_write(struct cgroup *cont, struct cftype *cft,
+    char *buffer)
{
    return res_counter_write(&mem_cgroup_from_cont(cont)->res,
-    cft->private, userbuf, nbytes, ppos,
-    mem_cgroup_write_strategy);
+    cft->private, buffer,
+    mem_cgroup_write_strategy);
}

static int mem_cgroup_reset(struct cgroup *cont, unsigned int event)
@@@ -929,7 +928,7 @@ static struct cftype mem_cgroup_files[]
{
    .name = "limit_in_bytes",
    .private = RES_LIMIT,
-    .write = mem_cgroup_write,
+    .write_string = mem_cgroup_write,
    .read_u64 = mem_cgroup_read,
},
{
--

```

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