Subject: [PATCH 8/8] CGroup Files: Convert res_counter_write() to be a cgroups write_string() handler
Posted by Paul Menage on Fri, 20 Jun 2008 23:44:06 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 simplies 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 | 11 +++++++---
 kernel/res_counter.c | 47 ++++++++++++++++++++++++++-------------------
 mm/memcontrol.c | 24 ++++++++++++++----------
 mm/memrlimitcgroup.c | 23 ++++++++++++++---------
 4 files changed, 38 insertions(+), 67 deletions(-)

Index: cws-2.6.26-rc5-mm3/include/linux/res_counter.h
===
--- cws-2.6.26-rc5-mm3.orig/include/linux/res_counter.h
+++ cws-2.6.26-rc5-mm3/include/linux/res_counter.h
@@ -63,9 +63,14 @@ 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 (*write_strategy)(char *buf, unsigned long long *val));
+
+typedef int (*write_strategy_fn)(const char *buf, unsigned long long *val);
+
+int res_counter_memparse_write_strategy(const char *buf,
+   unsigned long long *res);
+
+int res_counter_write(struct res_counter *counter, int member,
+   const char *buffer, write_strategy_fn write_strategy);

 /*
  * the field descriptors. one for each member of res_counter
Index: cws-2.6.26-rc5-mm3/kernel/res_counter.c
===
--- cws-2.6.26-rc5-mm3.orig/kernel/res_counter.c

```
+++ cws-2.6.26-rc5-mm3/kernel/res_counter.c
@@ -102,44 +102,37 @@ 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_memparse_write_strategy(const char *buf,
+      unsigned long long *res)
 {
- int ret;
- char *buf, *end;
- unsigned long flags;
- unsigned long long tmp, *val;
+ char *end;
+ /* FIXME - make memparse() take const char* args */
+ *res = memparse((char *)buf, &end);
+ if (*end != '\0')
+  return -EINVAL;

- 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;
+ *res = PAGE_ALIGN(*res);
+ return 0;
+}

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

- 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: cws-2.6.26-rc5-mm3/mm/memcontrol.c
===================================================================
--- cws-2.6.26-rc5-mm3.orig/mm/memcontrol.c
+++ cws-2.6.26-rc5-mm3/mm/memcontrol.c
@@ -853,32 +853,18 @@ out:
  return ret;
 }

-static int mem_cgroup_write_strategy(char *buf, unsigned long long *tmp)
-{
- *tmp = memparse(buf, &buf);
- if (*buf != '\0')
-  return -EINVAL;
-
- /*
-  * Round up the value to the closest page size
-  */
- *tmp = ((*tmp + PAGE_SIZE - 1) >> PAGE_SHIFT) << PAGE_SHIFT;
- return 0;
-}
-
 static u64 mem_cgroup_read(struct cgroup *cont, struct cftype *cft)
 {
  return res_counter_read_u64(&mem_cgroup_from_cont(cont)->res,
        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,
+       const char *buffer)
 {
  return res_counter_write(&mem_cgroup_from_cont(cont)->res,
-    cft->private, userbuf, nbytes, ppos,
-    mem_cgroup_write_strategy);
+     cft->private, buffer,
+     res_counter_memparse_write_strategy);
 }

 static int mem_cgroup_reset(struct cgroup *cont, unsigned int event)
@@ -968,7 +954,7 @@ 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,
  },
  {
Index: cws-2.6.26-rc5-mm3/mm/memrlimitcgroup.c
===================================================================
--- cws-2.6.26-rc5-mm3.orig/mm/memrlimitcgroup.c
+++ cws-2.6.26-rc5-mm3/mm/memrlimitcgroup.c
@@ -118,25 +118,12 @@ static u64 memrlimit_cgroup_read(struct
      cft->private);
 }

-static int memrlimit_cgroup_write_strategy(char *buf, unsigned long long *tmp)
-{
- *tmp = memparse(buf, &buf);
- if (*buf != '\0')
-  return -EINVAL;
-
- *tmp = PAGE_ALIGN(*tmp);
- return 0;
-}
-
-static ssize_t memrlimit_cgroup_write(struct cgroup *cgrp, struct cftype *cft,
-     struct file *file,
-     const char __user *userbuf,
-     size_t nbytes,
-     loff_t *ppos)
+static int memrlimit_cgroup_write(struct cgroup *cgrp, struct cftype *cft,
+       const char *buffer)
 {
  return res_counter_write(&memrlimit_cgroup_from_cgrp(cgrp)->as_res,
-     cft->private, userbuf, nbytes, ppos,
```

```
-     memrlimit_cgroup_write_strategy);
+     cft->private, buffer,
+     res_counter_memparse_write_strategy);
 }

 static struct cftype memrlimit_cgroup_files[] = {
@@ -148,7 +135,7 @@ static struct cftype memrlimit_cgroup_fi
  {
   .name = "limit_in_bytes",
   .private = RES_LIMIT,
-  .write = memrlimit_cgroup_write,
+  .write_string = memrlimit_cgroup_write,
   .read_u64 = memrlimit_cgroup_read,
  },
  {

--
```

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