Subject: Re: [PATCH] Add a 'trigger' callback on struct cftype.
Posted by Pavel Emelianov on Tue, 11 Mar 2008 16:13:12 GMT
View Forum Message <> Reply to Message

Paul Menage wrote:
> On Tue, Mar 11, 2008 at 8:21 AM, Pavel Emelyanov <xemul@openvz.org> wrote:
>> If the patch with max_usage for res_counter will be accepted we'll have
>> two :) files, that a event-triggers essentially, i.e. they don't care
>> what the user actually write to then, but are interested in the writing
>> by its own.
>>
>> So the proposal is to make cgroups infrastructure handle this case.
>
> This could be useful, but in the case of force_empty don't we lose the
> ability to report an error (EBUSY?) in the event that the cgroup still
> has tasks?

Yikes :(Good catch. The fix, however, is pretty small.

```
diff --git a/include/linux/cgroup.h b/include/linux/cgroup.h
index df579e3..f6b882d 100644
--- a/include/linux/cgroup.h
+++ b/include/linux/cgroup.h
@@ -243,7 +243,7 @@ struct cftype {
         */
        int (*write_s64) (struct cgroup *cgrp, struct cftype *cft, s64 val);

-       void (*trigger) (struct cgroup *cgrp, unsigned int event);
+       int (*trigger) (struct cgroup *cgrp, unsigned int event);
        int (*release) (struct inode *inode, struct file *file);
 };

diff --git a/kernel/cgroup.c b/kernel/cgroup.c
index 7d73c2b..f2d8f25 100644
--- a/kernel/cgroup.c
+++ b/kernel/cgroup.c
@@ -1411,8 +1411,8 @@ static ssize_t cgroup_file_write(struct file *file, const char __user *buf,
        if (cft->write_u64 || cft->write_s64)
                return cgroup_write_X64(cgrp, cft, file, buf, nbytes, ppos);
        if (cft->trigger) {
-               cft->trigger(cgrp, (unsigned int)cft->private);
-               return nbytes;
+               int ret = cft->trigger(cgrp, (unsigned int)cft->private);
+               return ret ? ret : nbytes;
        }
        return -EINVAL;
 }
diff --git a/mm/memcontrol.c b/mm/memcontrol.c
```

```
index 4c1d24c..ab1a862 100644
--- a/mm/memcontrol.c
+++ b/mm/memcontrol.c
@@ -868,14 +868,15 @@ static ssize_t mem_cgroup_write(struct cgroup *cont, struct cftype *cft,
                                mem_cgroup_write_strategy);
 }

-static void mem_cgroup_max_reset(struct cgroup *cont, unsigned int event)
+static int mem_cgroup_max_reset(struct cgroup *cont, unsigned int event)
 {
        res_counter_reset_max(&mem_cgroup_from_cont(cont)->res);
+       return 0;
 }

-static void mem_force_empty_write(struct cgroup *cont, unsigned int event)
+static int mem_force_empty_write(struct cgroup *cont, unsigned int event)
 {
-       mem_cgroup_force_empty(mem_cgroup_from_cont(cont));
+       return mem_cgroup_force_empty(mem_cgroup_from_cont(cont));
 }

 static const struct mem_cgroup_stat_desc {
```

> Paul
>