

---

Subject: [PATCH 0/3] Implement triggers for control groups.  
Posted by [Pavel Emelianov](#) on Thu, 13 Mar 2008 11:34:39 GMT  
[View Forum Message](#) <> [Reply to Message](#)

---

Hi, guys.

This is the final set I'm going to send to Andrew. It has a new feature - fail counter reset - and splitted, so I send it for a final review (maybe some Acked-by-s ;) ) before sending to Andrew.

Signed-off-by: Pavel Emelyanov <xemul@openvz.org>

---

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

---

---

Subject: [PATCH 1/3] Add the trigger callback to struct cftype  
Posted by [Pavel Emelianov](#) on Thu, 13 Mar 2008 11:36:17 GMT  
[View Forum Message](#) <> [Reply to Message](#)

---

Trigger callback can be used to receive a kick-up from the user space. The string written is ignored.

The cftype->private is used for multiplexing events.

Signed-off-by: Pavel Emelyanov <xemul@openvz.org>

---

```
include/linux/cgroup.h | 8 ++++++++
kernel/cgroup.c        | 4 ++++
2 files changed, 12 insertions(+), 0 deletions(-)
```

```
diff --git a/include/linux/cgroup.h b/include/linux/cgroup.h
index 785a01c..2d1d151 100644
--- a/include/linux/cgroup.h
+++ b/include/linux/cgroup.h
@@ -243,6 +243,14 @@ struct cftype {
    */
    int (*write_s64)(struct cgroup *cgrp, struct cftype *cft, s64 val);

+ /*
+  * trigger() callback can be used to get some kick from the
+  * userspace, when the actual string written is not important
+  * at all. The private field can be used to determine the
+  * kick type for multiplexing.
+  */
+ 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 e8e8ec4..f2d8f25 100644
--- a/kernel/cgroup.c
+++ b/kernel/cgroup.c
@@ -1410,6 +1410,10 @@ static ssize_t cgroup_file_write(struct file *file, const char __user *buf,
+ return cft->write(cgrp, cft, file, buf, nbytes, ppos);
+ if (cft->write_u64 || cft->write_s64)
+ return cgroup_write_X64(cgrp, cft, file, buf, nbytes, ppos);
+ if (cft->trigger) {
+ int ret = cft->trigger(cgrp, (unsigned int)cft->private);
+ return ret ? ret : nbytes;
+ }
+ return -EINVAL;
+ }

--
1.5.3.4

```

---

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

---



---

Subject: [PATCH 2/3] Use triggers in force\_empty and max\_usage files  
Posted by [Pavel Emelianov](#) on Thu, 13 Mar 2008 11:37:23 GMT  
[View Forum Message](#) <> [Reply to Message](#)

---

Signed-off-by: Pavel Emelyanov <xemul@openvz.org>

```

---
mm/memcontrol.c | 21 ++++++-----
1 files changed, 6 insertions(+), 15 deletions(-)

diff --git a/mm/memcontrol.c b/mm/memcontrol.c
index c27141d..d3ec3e3 100644
--- a/mm/memcontrol.c
+++ b/mm/memcontrol.c
@@ -868,27 +868,18 @@ static ssize_t mem_cgroup_write(struct cgroup *cont, struct cftype *cft,
+ mem_cgroup_write_strategy);
+ }

-static ssize_t mem_cgroup_max_reset(struct cgroup *cont, struct cftype *cft,
- struct file *file, const char __user *userbuf,

```

```

-   size_t nbytes, loff_t *ppos)
+static int mem_cgroup_max_reset(struct cgroup *cont, unsigned int event)
{
    struct mem_cgroup *mem;

    mem = mem_cgroup_from_cont(cont);
    res_counter_reset_max(&mem->res);
- return nbytes;
+ return 0;
}

-static ssize_t mem_force_empty_write(struct cgroup *cont,
-   struct cftype *cft, struct file *file,
-   const char __user *userbuf,
-   size_t nbytes, loff_t *ppos)
+static int mem_force_empty_write(struct cgroup *cont, unsigned int event)
{
- struct mem_cgroup *mem = mem_cgroup_from_cont(cont);
- int ret = mem_cgroup_force_empty(mem);
- if (!ret)
-   ret = nbytes;
- return ret;
+ return mem_cgroup_force_empty(mem_cgroup_from_cont(cont));
}

static const struct mem_cgroup_stat_desc {
@@ -936,7 +927,7 @@ static struct cftype mem_cgroup_files[] = {
{
    .name = "max_usage_in_bytes",
    .private = RES_MAX_USAGE,
-   .write = mem_cgroup_max_reset,
+   .trigger = mem_cgroup_max_reset,
    .read_u64 = mem_cgroup_read,
},
{
@@ -952,7 +943,7 @@ static struct cftype mem_cgroup_files[] = {
},
{
    .name = "force_empty",
-   .write = mem_force_empty_write,
+   .trigger = mem_force_empty_write,
},
{
    .name = "stat",
--
1.5.3.4

```

---

Subject: [PATCH 3/3] Implement failcounter reset  
Posted by [Pavel Emelianov](#) on Thu, 13 Mar 2008 11:39:55 GMT  
[View Forum Message](#) <> [Reply to Message](#)

---

Merge two resets into one mem\_cgroup\_reset() function to demonstrate how multiplexing work.

Besides, I have plans to move the files, that correspond to res\_counter to the res\_counter.c file and somehow "import" them into controller. I don't know how to make it gracefully yet, but merging resets of max\_usage and failcnt in one function will be there for sure.

Signed-off-by: Pavel Emelyanov <xemul@openvz.org>

```
---
include/linux/res_counter.h | 8 ++++++++
mm/memcontrol.c             | 14 ++++++++-----
2 files changed, 19 insertions(+), 3 deletions(-)

diff --git a/include/linux/res_counter.h b/include/linux/res_counter.h
index df8085a..3c05d6d 100644
--- a/include/linux/res_counter.h
+++ b/include/linux/res_counter.h
@@ -141,4 +141,12 @@ static inline void res_counter_reset_max(struct res_counter *cnt)
    spin_unlock_irqrestore(&cnt->lock, flags);
}

+static inline void res_counter_reset_failcnt(struct res_counter *cnt)
+{
+ unsigned long flags;
+
+ spin_lock_irqsave(&cnt->lock, flags);
+ cnt->failcnt = 0;
+ spin_unlock_irqrestore(&cnt->lock, flags);
+}
+
#endif
diff --git a/mm/memcontrol.c b/mm/memcontrol.c
index d3ec3e3..5852b23 100644
--- a/mm/memcontrol.c
+++ b/mm/memcontrol.c
@@ -868,12 +868,19 @@ static ssize_t mem_cgroup_write(struct cgroup *cont, struct cftype *cft,
    mem_cgroup_write_strategy);
```

```
}
```

```
-static int mem_cgroup_max_reset(struct cgroup *cont, unsigned int event)
```

```
+static int mem_cgroup_reset(struct cgroup *cont, unsigned int event)
```

```
{  
    struct mem_cgroup *mem;  
  
    mem = mem_cgroup_from_cont(cont);  
- res_counter_reset_max(&mem->res);  
+ switch (event) {  
+ case RES_MAX_USAGE:  
+   res_counter_reset_max(&mem->res);  
+   break;  
+ case RES_FAILCNT:  
+   res_counter_reset_failcnt(&mem->res);  
+   break;  
+ }  
    return 0;  
}
```

```
@ @ -927,7 +934,7 @ @ static struct cftype mem_cgroup_files[] = {
```

```
{  
    .name = "max_usage_in_bytes",  
    .private = RES_MAX_USAGE,  
- .trigger = mem_cgroup_max_reset,  
+ .trigger = mem_cgroup_reset,  
    .read_u64 = mem_cgroup_read,  
},  
{
```

```
@ @ -939,6 +946,7 @ @ static struct cftype mem_cgroup_files[] = {
```

```
{  
    .name = "failcnt",  
    .private = RES_FAILCNT,  
+ .trigger = mem_cgroup_reset,  
    .read_u64 = mem_cgroup_read,  
},  
{
```

```
--
```

1.5.3.4

---

Containers mailing list

Containers@lists.linux-foundation.org

<https://lists.linux-foundation.org/mailman/listinfo/containers>

---

---

Subject: Re: [PATCH 1/3] Add the trigger callback to struct cftype

---

Posted by [Paul Menage](#) on Thu, 13 Mar 2008 12:19:04 GMT

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

---

On Thu, Mar 13, 2008 at 4:36 AM, Pavel Emelyanov <xemul@openvz.org> wrote:

```
> Trigger callback can be used to receive a kick-up from the
> user space. The string written is ignored.
>
> The cftype->private is used for multiplexing events.
>
> Signed-off-by: Pavel Emelyanov <xemul@openvz.org>
```

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

I'm not sure about the behaviour of passing cft->private rather than just cft, but we can always change that later if it turns that some user needs the cft pointer for some other reason.

Paul

```
>
> ---
> include/linux/cgroup.h | 8 ++++++++
> kernel/cgroup.c        | 4 ++++
> 2 files changed, 12 insertions(+), 0 deletions(-)
>
> diff --git a/include/linux/cgroup.h b/include/linux/cgroup.h
> index 785a01c..2d1d151 100644
> --- a/include/linux/cgroup.h
> +++ b/include/linux/cgroup.h
> @@ -243,6 +243,14 @@ struct cftype {
>     */
>     int (*write_s64) (struct cgroup *cgrp, struct cftype *cft, s64 val);
>
> +     /*
> +      * trigger() callback can be used to get some kick from the
> +      * userspace, when the actual string written is not important
> +      * at all. The private field can be used to determine the
> +      * kick type for multiplexing.
> +      */
> +     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 e8e8ec4..f2d8f25 100644
> --- a/kernel/cgroup.c
> +++ b/kernel/cgroup.c
> @@ -1410,6 +1410,10 @@ static ssize_t cgroup_file_write(struct file *file, const char __user
```

```
*buf,  
>         return cft->write(cgrp, cft, file, buf, nbytes, ppos);  
>     if (cft->write_u64 || cft->write_s64)  
>         return cgroup_write_X64(cgrp, cft, file, buf, nbytes, ppos);  
> +     if (cft->trigger) {  
> +         int ret = cft->trigger(cgrp, (unsigned int)cft->private);  
> +         return ret ? ret : nbytes;  
> +     }  
>     return -EINVAL;  
> }  
>  
> --  
> 1.5.3.4  
>  
>
```

---

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

---

---

Subject: Re: [PATCH 0/3] Implement triggers for control groups.  
Posted by [KAMEZAWA Hiroyuki](#) on Fri, 14 Mar 2008 02:08:23 GMT  
[View Forum Message](#) <> [Reply to Message](#)

---

On Thu, 13 Mar 2008 14:34:39 +0300  
Pavel Emelyanov <xemul@openvz.org> wrote:

```
> Hi, guys.  
>  
> This is the final set I'm going to send to Andrew. It has a new  
> feature - fail counter reset - and splitted, so I send it for a  
> final review (maybe some Aacked-by-s ;) ) before sending to Andrew.  
>  
> Signed-off-by: Pavel Emelyanov <xemul@openvz.org>
```

I like this new interface. thank you.

Acked-by: KAMEZAWA Hiroyuki <kamezawa.hiroyu@jp.fujitsu.com>

---

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

---