
Subject: [PATCH] cgroups: respect cgroup timer slack value in prctl

Posted by [jacob.jun.pan](#) on Tue, 01 Feb 2011 22:41:43 GMT

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

From: Jacob Pan <jacob.jun.pan@linux.intel.com>

Per cgroup timer slack value is used to manage task groups for more friendly power management behavior. Once a task is attached to a cgroup, we should not allow user to change (escape) the per cgroup restriction unless a more relaxed timer slack value is given by user via prctl.

Signed-off-by: Jacob Pan <jacob.jun.pan@linux.intel.com>

```
include/linux/cgroup.h    | 13 ++++++++
kernel/cgroup_timer_slack.c |  4 ----
kernel/sys.c              | 21 ++++++++
3 files changed, 33 insertions(+), 5 deletions(-)
```

diff --git a/include/linux/cgroup.h b/include/linux/cgroup.h

index ed4ba11..4405666 100644

--- a/include/linux/cgroup.h

+++ b/include/linux/cgroup.h

```
@@ -627,6 +627,19 @@ bool css_is_ancestor(struct cgroup_subsys_state *cg,
unsigned short css_id(struct cgroup_subsys_state *css);
unsigned short css_depth(struct cgroup_subsys_state *css);
```

```
+#ifdef CONFIG_CGROUP_TIMER_SLACK
```

```
+struct timer_slack_cgroup {
```

```
+ struct cgroup_subsys_state css;
```

```
+ unsigned long timer_slack_ns;
```

```
+};
```

```
+
```

```
+static inline struct timer_slack_cgroup *task_timer_slack(struct task_struct *task)
```

```
+{
```

```
+ return container_of(task_subsys_state(task, timer_slack_subsys_id),
```

```
+     struct timer_slack_cgroup, css);
```

```
+}
```

```
+#endif
```

```
+
```

```
#else /* !CONFIG_CGROUPS */
```

```
static inline int cgroup_init_early(void) { return 0; }
```

diff --git a/kernel/cgroup_timer_slack.c b/kernel/cgroup_timer_slack.c

index daa452d..1de122a 100644

--- a/kernel/cgroup_timer_slack.c

+++ b/kernel/cgroup_timer_slack.c

```
@@ -4,10 +4,6 @@
```

```
#include <linux/slab.h>
```

```
struct cgroup_subsys timer_slack_subsys;
```

```
-struct timer_slack_cgroup {  
- struct cgroup_subsys_state css;  
- unsigned long timer_slack_ns;  
-};
```

```
static struct timer_slack_cgroup *cgroup_to_tslack_cgroup(struct cgroup *cgroup)  
{
```

```
diff --git a/kernel/sys.c b/kernel/sys.c
```

```
index 7f5a0cd..433fdb 100644
```

```
--- a/kernel/sys.c
```

```
+++ b/kernel/sys.c
```

```
@@ -37,6 +37,8 @@
```

```
#include <linux/ptrace.h>
```

```
#include <linux/fs_struct.h>
```

```
#include <linux/gfp.h>
```

```
+#include <linux/cgroup.h>
```

```
+#include <linux/init_task.h>
```

```
#include <linux/compat.h>
```

```
#include <linux/syscalls.h>
```

```
@@ -1688,8 +1690,25 @@ SYSCALL_DEFINE5(prctl, int, option, unsigned long, arg2, unsigned  
long, arg3,
```

```
    if (arg2 <= 0)
```

```
        current->timer_slack_ns =
```

```
        current->default_timer_slack_ns;
```

```
- else
```

```
+ else {
```

```
+#ifdef CONFIG_CGROUP_TIMER_SLACK
```

```
+ /* only allow more relaxed slack value than
```

```
+ * the attached timer slack cgroup allows.
```

```
+ */
```

```
+ unsigned long cg_timer_slack;
```

```
+
```

```
+ rcu_read_lock();
```

```
+ cg_timer_slack = task_timer_slack(current)->timer_slack_ns;
```

```
+ rcu_read_unlock();
```

```
+ if (cg_timer_slack < arg2 || cg_timer_slack
```

```
+ == TIMER_SLACK_NS_DEFAULT)
```

```
+    current->timer_slack_ns = arg2;
```

```
+ else
```

```
+    return -EINVAL;
```

```
+#else
```

```
    current->timer_slack_ns = arg2;
```

```
+#endif
```

```
+ }
```

```
error = 0;  
break;  
case PR_MCE_KILL:
```

```
--
```

1.7.0.4

Containers mailing list

Containers@lists.linux-foundation.org

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