
Subject: [PATCH 13/29] task containersv11 simple task container debug info subsystem

Posted by [Paul Menage](#) on Tue, 11 Sep 2007 19:52:52 GMT

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

From: Paul Menage <menage@google.com>

This example subsystem exports debugging information as an aid to diagnosing refcount leaks, etc, in the cgroup framework.

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

Cc: Serge E. Hallyn <serue@us.ibm.com>

Cc: "Eric W. Biederman" <ebiederm@xmission.com>

Cc: Dave Hansen <haveblue@us.ibm.com>

Cc: Balbir Singh <balbir@in.ibm.com>

Cc: Paul Jackson <pj@sgi.com>

Cc: Kirill Korotaev <dev@openvz.org>

Cc: Herbert Poetzl <herbert@13thfloor.at>

Cc: Srivatsa Vaddagiri <vatsa@in.ibm.com>

Cc: Cedric Le Goater <clg@fr.ibm.com>

Signed-off-by: Andrew Morton <akpm@linux-foundation.org>

```
include/linux/cgroup_subsys.h | 4 +
init/Kconfig                  | 10 ++
kernel/Makefile               | 1
kernel/cgroup_debug.c         | 97 +++++
4 files changed, 112 insertions(+)
```

diff -puN

include/linux/cgroup_subsys.h~task-cgroupsv11-simple-task-cgroup-debug-info-subsystem

include/linux/cgroup_subsys.h

--- a/include/linux/cgroup_subsys.h~task-cgroupsv11-simple-task-cgroup-debug-info-subsystem

+++ a/include/linux/cgroup_subsys.h

@@ -19,4 +19,8 @@ SUBSYS(cpuacct)

/* */

+#ifdef CONFIG_CGROUP_DEBUG

+SUBSYS(debug)

+#endif

+

/* */

diff -puN init/Kconfig~task-cgroupsv11-simple-task-cgroup-debug-info-subsystem init/Kconfig

--- a/init/Kconfig~task-cgroupsv11-simple-task-cgroup-debug-info-subsystem

+++ a/init/Kconfig

@@ -282,6 +282,16 @@ config CGROUPS

Say N if unsure.

```
+config CGROUP_DEBUG
+ bool "Example debug cgroup subsystem"
+ depends on CGROUPS
+ help
+   This option enables a simple cgroup subsystem that
+   exports useful debugging information about the cgroups
+   framework
+
+   Say N if unsure
+
config CPUSETS
  bool "Cpuset support"
  depends on SMP && CGROUPS
diff -puN kernel/Makefile~task-cgroupsv11-simple-task-cgroup-debug-info-subsystem
kernel/Makefile
--- a/kernel/Makefile~task-cgroupsv11-simple-task-cgroup-debug-info-subsystem
+++ a/kernel/Makefile
@@ -39,6 +39,7 @@ obj-$(CONFIG_BSD_PROCESS_ACCT) += acct.o
obj-$(CONFIG_KEXEC) += kexec.o
obj-$(CONFIG_COMPAT) += compat.o
obj-$(CONFIG_CGROUPS) += cgroup.o
+obj-$(CONFIG_CGROUP_DEBUG) += cgroup_debug.o
obj-$(CONFIG_CPUSETS) += cpuset.o
obj-$(CONFIG_CGROUP_CPUACCT) += cpu_acct.o
obj-$(CONFIG_IKCONFIG) += configs.o
diff -puN /dev/null kernel/cgroup_debug.c
--- /dev/null
+++ a/kernel/cgroup_debug.c
@@ -0,0 +1,97 @@
+/*
+ * kernel/ccontainer_debug.c - Example cgroup subsystem that
+ * exposes debug info
+ *
+ * Copyright (C) Google Inc, 2007
+ *
+ * Developed by Paul Menage (menage@google.com)
+ */
+
+#include <linux/cgroup.h>
+#include <linux/fs.h>
+#include <linux/slab.h>
+#include <linux/rcupdate.h>
+
+#include <asm/atomic.h>
```

```

+static struct cgroup_subsys_state *debug_create(struct cgroup_subsys *ss,
+        struct cgroup *cont)
+{
+ struct cgroup_subsys_state *css = kzalloc(sizeof(*css), GFP_KERNEL);
+
+ if (!css)
+ return ERR_PTR(-ENOMEM);
+
+ return css;
+}
+
+static void debug_destroy(struct cgroup_subsys *ss, struct cgroup *cont)
+{
+ kfree(cont->subsys[debug_subsys_id]);
+}
+
+static u64 cgroup_refcount_read(struct cgroup *cont, struct cftype *cft)
+{
+ return atomic_read(&cont->count);
+}
+
+static u64 taskcount_read(struct cgroup *cont, struct cftype *cft)
+{
+ u64 count;
+
+ cgroup_lock();
+ count = cgroup_task_count(cont);
+ cgroup_unlock();
+ return count;
+}
+
+static u64 current_css_set_read(struct cgroup *cont, struct cftype *cft)
+{
+ return (u64)(long)current->cgroups;
+}
+
+static u64 current_css_set_refcount_read(struct cgroup *cont,
+        struct cftype *cft)
+{
+ u64 count;
+
+ rcu_read_lock();
+ count = atomic_read(&current->cgroups->ref.refcount);
+ rcu_read_unlock();
+ return count;
+}
+
+static struct cftype files[] = {

```

```
+ {
+ .name = "cgroup_refcount",
+ .read_uint = cgroup_refcount_read,
+ },
+ {
+ .name = "taskcount",
+ .read_uint = taskcount_read,
+ },
+
+ {
+ .name = "current_css_set",
+ .read_uint = current_css_set_read,
+ },
+
+ {
+ .name = "current_css_set_refcount",
+ .read_uint = current_css_set_refcount_read,
+ },
+ };
+
+static int debug_populate(struct cgroup_subsys *ss, struct cgroup *cont)
+{
+ return cgroup_add_files(cont, ss, files, ARRAY_SIZE(files));
+}
+
+struct cgroup_subsys debug_subsys = {
+ .name = "debug",
+ .create = debug_create,
+ .destroy = debug_destroy,
+ .populate = debug_populate,
+ .subsys_id = debug_subsys_id,
+ };
+
--
```