Subject: [PATCH 29/33] memory controller oom handling v7
Posted by Paul Menage on Mon, 17 Sep 2007 21:03:36 GMT
View Forum Message <> Reply to Message

From: Pavel Emelianov <xemul@openvz.org>
(container->cgroup renaming by Paul Menage <menage@google.com>)

Out of memory handling for cgroups over their limit. A task from the
cgroup over limit is chosen using the existing OOM logic and killed.

TODO:
1. As discussed in the OLS BOF session, consider implementing a user
space policy for OOM handling.

Signed-off-by: Pavel Emelianov <xemul@openvz.org>
Signed-off-by: Balbir Singh <balbir@linux.vnet.ibm.com>
Signed-off-by: Paul Menage <menage@google.com>
---

 include/linux/memcontrol.h | 1
 mm/memcontrol.c | 1
 mm/oom_kill.c | 42 ++++++++++++++++++++++++++++++++++++++----
 3 files changed, 40 insertions(+), 4 deletions(-)

diff -puN include/linux/memcontrol.h~memory-controller-oom-handling-v7
include/linux/memcontrol.h
--- a/include/linux/memcontrol.h~memory-controller-oom-handling-v7
+++ a/include/linux/memcontrol.h
@@ -39,6 +39,7 @@ extern unsigned long mem_cgroup_isola
      int mode, struct zone *z,
      struct mem_cgroup *mem_cont,
      int active);
+extern void mem_cgroup_out_of_memory(struct mem_cgroup *mem);

 static inline void mem_cgroup_uncharge_page(struct page *page)
 {
diff -puN mm/memcontrol.c~memory-controller-oom-handling-v7 mm/memcontrol.c
--- a/mm/memcontrol.c~memory-controller-oom-handling-v7
+++ a/mm/memcontrol.c
@@ -322,6 +322,7 @@ int mem_cgroup_charge(struct page *pa
   }

   css_put(&mem->css);
+ mem_cgroup_out_of_memory(mem);
   goto free_pc;
  }

diff -puN mm/oom_kill.c~memory-controller-oom-handling-v7 mm/oom_kill.c

```
--- a/mm/oom_kill.c~memory-controller-oom-handling-v7
+++ a/mm/oom_kill.c
@@ -25,6 +25,7 @@
 #include <linux/cpuset.h>
 #include <linux/module.h>
 #include <linux/notifier.h>
+#include <linux/memcontrol.h>

 int sysctl_panic_on_oom;
 /* #define DEBUG */
@@ -48,7 +49,8 @@ int sysctl_panic_on_oom;
  *    of least surprise ... (be careful when you change it)
  */

-unsigned long badness(struct task_struct *p, unsigned long uptime)
+unsigned long badness(struct task_struct *p, unsigned long uptime,
+   struct mem_cgroup *mem)
 {
  unsigned long points, cpu_time, run_time, s;
  struct mm_struct *mm;
@@ -61,6 +63,13 @@ unsigned long badness(struct task_struct
   return 0;
  }

+#ifdef CONFIG_CGROUP_MEM_CONT
+ if (mem != NULL && mm->mem_cgroup != mem) {
+  task_unlock(p);
+  return 0;
+ }
+#endif
+
  /*
   * The memory size of the process is the basis for the badness.
   */
@@ -198,7 +207,8 @@ static inline int constrained_alloc(stru
  *
  * (not docbooked, we don't want this one cluttering up the manual)
  */
-static struct task_struct *select_bad_process(unsigned long *ppoints)
+static struct task_struct *select_bad_process(unsigned long *ppoints,
+      struct mem_cgroup *mem)
 {
  struct task_struct *g, *p;
  struct task_struct *chosen = NULL;
@@ -252,7 +262,7 @@ static struct task_struct *select_bad_pr
   if (p->oomkilladj == OOM_DISABLE)
    continue;
```

```
-  points = badness(p, uptime.tv_sec);
+  points = badness(p, uptime.tv_sec, mem);
   if (points > *ppoints || !chosen) {
    chosen = p;
    *ppoints = points;
@@ -364,6 +374,30 @@ static int oom_kill_process(struct task_
  return oom_kill_task(p);
 }

+#ifdef CONFIG_CGROUP_MEM_CONT
+void mem_cgroup_out_of_memory(struct mem_cgroup *mem)
+{
+ unsigned long points = 0;
+ struct task_struct *p;
+
+ cgroup_lock();
+ rcu_read_lock();
+retry:
+ p = select_bad_process(&points, mem);
+ if (PTR_ERR(p) == -1UL)
+  goto out;
+
+ if (!p)
+  p = current;
+
+ if (oom_kill_process(p, points, "Memory cgroup out of memory"))
+  goto retry;
+out:
+ rcu_read_unlock();
+ cgroup_unlock();
+}
+#endif
+
 static BLOCKING_NOTIFIER_HEAD(oom_notify_list);

 int register_oom_notifier(struct notifier_block *nb)
@@ -436,7 +470,7 @@ retry:
    * Rambo mode: Shoot down a process and hope it solves whatever
    * issues we may have.
    */
-  p = select_bad_process(&points);
+  p = select_bad_process(&points, NULL);

   if (PTR_ERR(p) == -1UL)
    goto out;
_

--
```

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