Subject: [PATCH 16/20] Create a slab-cache for 'struct pid_namespace'
Posted by Pavel Emelianov on Fri, 10 Aug 2007 11:48:22 GMT
View Forum Message <> Reply to Message

From: Sukadev Bhattiprolu <sukadev@us.ibm.com>

This will help fixing memory leaks due to bad reference counting.

Signed-off-by: Sukadev Bhattiprolu <sukadev@us.ibm.com>
Cc: Oleg Nesterov <oleg@tv-sign.ru>

---

 pid.c | 9 ++++++---
 1 files changed, 6 insertions(+), 3 deletions(-)

--- ./kernel/pid.c.ve16 2007-08-07 12:50:16.000000000 +0400
+++ ./kernel/pid.c 2007-08-07 12:51:13.000000000 +0400
@@ -34,6 +34,7 @@
 static struct hlist_head *pid_hash;
 static int pidhash_shift;
 struct pid init_struct_pid = INIT_STRUCT_PID;
+static struct kmem_cache *pid_ns_cachep;

 int pid_max = PID_MAX_DEFAULT;

@@ -480,7 +481,7 @@ static struct pid_namespace *create_pid_
  struct pid_namespace *ns;
  int i;

- ns = kmalloc(sizeof(struct pid_namespace), GFP_KERNEL);
+ ns = kmem_cache_alloc(pid_ns_cachep, GFP_KERNEL);
  if (ns == NULL)
   goto out;

@@ -510,7 +511,7 @@ static struct pid_namespace *create_pid_
 out_free_map:
  kfree(ns->pidmap[0].page);
 out_free:
- kfree(ns);
+ kmem_cache_free(pid_ns_cachep, ns);
 out:
  return ERR_PTR(-ENOMEM);
 }
@@ -521,7 +522,7 @@ static void destroy_pid_namespace(struct

  for (i = 0; i < PIDMAP_ENTRIES; i++)
   kfree(ns->pidmap[i].page);

```
- kfree(ns);
+ kmem_cache_free(pid_ns_cachep, ns);
 }

 struct pid_namespace *copy_pid_ns(unsigned long flags, struct pid_namespace *old_ns)
@@ -595,4 +596,6 @@ void __init pidmap_init(void)
  init_pid_ns.pid_cachep = create_pid_cachep(1);
  if (init_pid_ns.pid_cachep == NULL)
   panic("Can't create pid_1 cachep\n");
+
+ pid_ns_cachep = KMEM_CACHE(pid_namespace, SLAB_PANIC);
 }
```