Subject: [RFC][PATCH v2 2/7] Scaling msgmni to the number of ipc namespaces
Posted by Nadia Derbey on Thu, 31 Jan 2008 13:40:20 GMT
View Forum Message <> Reply to Message

[PATCH 02/07]

Since all the namespaces see the same amount of memory (the total one)
this patch introduces a new variable that counts the ipc namespaces and
divides msg_ctlmni by this counter.

Signed-off-by: Nadia Derbey <Nadia.Derbey@bull.net>

---
 include/linux/ipc.h | 1 +
 ipc/msg.c | 10 +++++++---
 ipc/util.c | 7 +++++++
 3 files changed, 15 insertions(+), 3 deletions(-)

Index: linux-2.6.24/include/linux/ipc.h
===
--- linux-2.6.24.orig/include/linux/ipc.h 2008-01-29 16:54:38.000000000 +0100
+++ linux-2.6.24/include/linux/ipc.h 2008-01-31 10:03:47.000000000 +0100
@@ -121,6 +121,7 @@ struct ipc_namespace {
 };

 extern struct ipc_namespace init_ipc_ns;
+extern atomic_t nr_ipc_ns;

 #ifdef CONFIG_SYSVIPC
 #define INIT_IPC_NS(ns) .ns = &init_ipc_ns,
Index: linux-2.6.24/ipc/util.c
===
--- linux-2.6.24.orig/ipc/util.c 2008-01-29 16:55:04.000000000 +0100
+++ linux-2.6.24/ipc/util.c 2008-01-31 10:05:58.000000000 +0100
@@ -51,6 +51,9 @@ struct ipc_namespace init_ipc_ns = {
  },
 };

+atomic_t nr_ipc_ns = ATOMIC_INIT(1);
+
+
 static struct ipc_namespace *clone_ipc_ns(struct ipc_namespace *old_ns)
 {
  int err;
@@ -61,6 +64,8 @@ static struct ipc_namespace *clone_ipc_n
  if (ns == NULL)
   goto err_mem;

```
+ atomic_inc(&nr_ipc_ns);
+
  err = sem_init_ns(ns);
  if (err)
   goto err_sem;
@@ -80,6 +85,7 @@ err_msg:
  sem_exit_ns(ns);
 err_sem:
  kfree(ns);
+ atomic_dec(&nr_ipc_ns);
 err_mem:
  return ERR_PTR(err);
 }
@@ -109,6 +115,7 @@ void free_ipc_ns(struct kref *kref)
  msg_exit_ns(ns);
  shm_exit_ns(ns);
  kfree(ns);
+ atomic_dec(&nr_ipc_ns);
 }

 /**
Index: linux-2.6.24/ipc/msg.c
===================================================================
--- linux-2.6.24.orig/ipc/msg.c 2008-01-31 09:58:25.000000000 +0100
+++ linux-2.6.24/ipc/msg.c 2008-01-31 10:08:49.000000000 +0100
@@ -83,24 +83,28 @@ static int sysvipc_msg_proc_show(struct
 /*
  * Scale msgmni with the available lowmem size: the memory dedicated to msg
  * queues should occupy at most 1/MSG_MEM_SCALE of lowmem.
- * This should be done staying within the (MSGMNI , IPCMNI) range.
+ * Also take into account the number of nsproxies created so far.
+ * This should be done staying within the (MSGMNI , IPCMNI/nr_ipc_ns) range.
  */
 static void recompute_msgmni(struct ipc_namespace *ns)
 {
  struct sysinfo i;
  unsigned long allowed;
+ int nb_ns;

  si_meminfo(&i);
  allowed = (((i.totalram - i.totalhigh) / MSG_MEM_SCALE) * i.mem_unit)
   / MSGMNB;
+ nb_ns = atomic_read(&nr_ipc_ns);
+ allowed /= nb_ns;

  if (allowed < MSGMNI) {
   ns->msg_ctlmni = MSGMNI;
   goto out_callback;
```

```
  }

- if (allowed > IPCMNI) {
-  ns->msg_ctlmni = IPCMNI;
+ if (allowed > IPCMNI / nb_ns) {
+  ns->msg_ctlmni = IPCMNI / nb_ns;
   goto out_callback;
  }


--
```

_____

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