
Subject: [PATCH 2/8] Scaling msgmni to the number of ipc namespaces

Posted by [Nadia Derby](#) on Mon, 11 Feb 2008 14:16:48 GMT

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

[PATCH 02/08]

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 Derby <Nadia.Derbey@bull.net>

```
---
include/linux/ipc_namespace.h | 1 +
ipc/msg.c                     | 10 ++++++---
ipc/namespaces.c              | 3 +++
ipc/util.c                    | 3 +++
4 files changed, 14 insertions(+), 3 deletions(-)
```

Index: linux-2.6.24-mm1/include/linux/ipc_namespace.h

```
=====
--- linux-2.6.24-mm1.orig/include/linux/ipc_namespace.h 2008-02-07 15:00:43.000000000 +0100
+++ linux-2.6.24-mm1/include/linux/ipc_namespace.h 2008-02-07 15:26:53.000000000 +0100
@@ -33,6 +33,7 @@ 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-mm1/ipc/util.c

```
=====
--- linux-2.6.24-mm1.orig/ipc/util.c 2008-02-07 13:41:07.000000000 +0100
+++ linux-2.6.24-mm1/ipc/util.c 2008-02-07 15:36:22.000000000 +0100
@@ -52,6 +52,9 @@ struct ipc_namespace init_ipc_ns = {
},
};
```

```
+atomic_t nr_ipc_ns = ATOMIC_INIT(1);
```

```
+
```

```
+
```

```
/**
```

```
 * ipc_init - initialise IPC subsystem
```

```
 *
```

Index: linux-2.6.24-mm1/ipc/namespaces.c

```
=====
--- linux-2.6.24-mm1.orig/ipc/namespaces.c 2008-02-07 13:41:07.000000000 +0100
```

```
+++ linux-2.6.24-mm1/ipc/namespace.c 2008-02-07 15:40:19.000000000 +0100
@@ -20,6 +20,8 @@ static struct ipc_namespace *clone_ipc_n
    if (ns == NULL)
        return ERR_PTR(-ENOMEM);
```

```
+ atomic_inc(&nr_ipc_ns);
+
```

```
    sem_init_ns(ns);
    msg_init_ns(ns);
    shm_init_ns(ns);
@@ -83,4 +85,5 @@ 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-mm1/ipc/msg.c

```
=====
```

```
--- linux-2.6.24-mm1.orig/ipc/msg.c 2008-02-07 15:24:19.000000000 +0100
+++ linux-2.6.24-mm1/ipc/msg.c 2008-02-07 15:43:51.000000000 +0100
@@ -82,24 +82,28 @@ 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>
