Subject: [PATCH 1/8] Scaling msgmni to the amount of lowmem
Posted by Nadia Derbey on Mon, 11 Feb 2008 14:16:47 GMT
View Forum Message <> Reply to Message

[PATCH 01/08]

This patch computes msg_ctlmni to make it scale with the amount of lowmem.
msg_ctlmni is now set to make the message queues occupy 1/32 of the available
lowmem.

Some cleaning has also been done for the MSGPOOL constant: the msgctl man page
says it's not used, but it also defines it as a size in bytes (the code
expresses it in Kbytes).

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

---
 include/linux/msg.h | 14 ++++++++++--
 ipc/msg.c | 37 ++++++++++++++++++++++++++++++++++++-
 2 files changed, 48 insertions(+), 3 deletions(-)

Index: linux-2.6.24-mm1/include/linux/msg.h
===
--- linux-2.6.24-mm1.orig/include/linux/msg.h 2008-02-07 15:01:38.000000000 +0100
+++ linux-2.6.24-mm1/include/linux/msg.h 2008-02-07 15:23:17.000000000 +0100
@@ -49,16 +49,26 @@ struct msginfo {
  unsigned short msgseg;
 };

+/*
+ * Scaling factor to compute msgmni:
+ * the memory dedicated to msg queues (msgmni * msgmnb) should occupy
+ * at most 1/MSG_MEM_SCALE of the lowmem (see the formula in ipc/msg.c):
+ * up to 8MB   : msgmni = 16 (MSGMNI)
+ * 4 GB   : msgmni = 8K
+ * more than 16 GB : msgmni = 32K (IPCMNI)
+ */
+#define MSG_MEM_SCALE 32
+
 #define MSGMNI   16   /* <= IPCMNI */   /* max # of msg queue identifiers */
 #define MSGMAX 8192   /* <= INT_MAX */   /* max size of message (bytes) */
 #define MSGMNB 16384   /* <= INT_MAX */   /* default max size of a message queue */

 /* unused */
-#define MSGPOOL (MSGMNI*MSGMNB/1024) /* size in kilobytes of message pool */
+#define MSGPOOL (MSGMNI * MSGMNB) /* size in bytes of message pool */
 #define MSGTQL MSGMNB   /* number of system message headers */
 #define MSGMAP MSGMNB   /* number of entries in message map */

```
 #define MSGSSZ  16              /* message segment size */
-#define __MSGSEG ((MSGPOOL*1024)/ MSGSSZ) /* max no. of segments */
+#define __MSGSEG (MSGPOOL / MSGSSZ) /* max no. of segments */
 #define MSGSEG (__MSGSEG <= 0xffff ? __MSGSEG : 0xffff)

 #ifdef __KERNEL__
Index: linux-2.6.24-mm1/ipc/msg.c
===================================================================
--- linux-2.6.24-mm1.orig/ipc/msg.c 2008-02-07 15:02:29.000000000 +0100
+++ linux-2.6.24-mm1/ipc/msg.c 2008-02-07 15:24:19.000000000 +0100
@@ -27,6 +27,7 @@
 #include <linux/msg.h>
 #include <linux/spinlock.h>
 #include <linux/init.h>
+#include <linux/mm.h>
 #include <linux/proc_fs.h>
 #include <linux/list.h>
 #include <linux/security.h>
@@ -78,11 +79,45 @@ static int newque(struct ipc_namespace *
 static int sysvipc_msg_proc_show(struct seq_file *s, void *it);
 #endif

+/*
+ * 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.
+ */
+static void recompute_msgmni(struct ipc_namespace *ns)
+{
+ struct sysinfo i;
+ unsigned long allowed;
+
+ si_meminfo(&i);
+ allowed = (((i.totalram - i.totalhigh) / MSG_MEM_SCALE) * i.mem_unit)
+  / MSGMNB;
+
+ if (allowed < MSGMNI) {
+  ns->msg_ctlmni = MSGMNI;
+  goto out_callback;
+ }
+
+ if (allowed > IPCMNI) {
+  ns->msg_ctlmni = IPCMNI;
+  goto out_callback;
+ }
+
+ ns->msg_ctlmni = allowed;
+
```

```
+out_callback:
+
+ printk(KERN_INFO "msgmni has been set to %d for ipc namespace %p\n",
+  ns->msg_ctlmni, ns);
+}
+
 void msg_init_ns(struct ipc_namespace *ns)
 {
  ns->msg_ctlmax = MSGMAX;
  ns->msg_ctlmnb = MSGMNB;
- ns->msg_ctlmni = MSGMNI;
+
+ recompute_msgmni(ns);
+
  atomic_set(&ns->msg_bytes, 0);
  atomic_set(&ns->msg_hdrs, 0);
  ipc_init_ids(&ns->ids[IPC_MSG_IDS]);

--
```

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