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]);

--
```

Subject: Re: [PATCH 1/8] Scaling msgmni to the amount of lowmem
Posted by akpm on Sat, 16 Feb 2008 05:59:16 GMT
View Forum Message <> Reply to Message

On Mon, 11 Feb 2008 15:16:47 +0100 Nadia.Derbey@bull.net wrote:

> [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).
>

Something's wrong here. Running LTP's msgctl08 (specifically:
ltp-full-20070228) cripples the machine. It's a 4-way 4GB x86_64.

http://userweb.kernel.org/~akpm/config-x.txt
http://userweb.kernel.org/~akpm/dmesg-x.txt

Normally msgctl08 will complete in a second or two. With this patch I
don't know how long it will take to complete, and the machine is horridly
bogged down. It does recover if you manage to kill msgctl08. Feels like
a terrible memory shortage, but there's plenty of memory free and it isn't
swapping.

Subject: Re: [PATCH 1/8] Scaling msgmni to the amount of lowmem
Posted by Nadia Derbey on Mon, 18 Feb 2008 09:19:08 GMT
View Forum Message <> Reply to Message

Andrew Morton wrote:
> On Mon, 11 Feb 2008 15:16:47 +0100 Nadia.Derbey@bull.net wrote:
>
>
>>[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).
>>
>
>
> Something's wrong here. Running LTP's msgctl08 (specifically:
> ltp-full-20070228) cripples the machine. It's a 4-way 4GB x86_64.
>
> http://userweb.kernel.org/~akpm/config-x.txt
> http://userweb.kernel.org/~akpm/dmesg-x.txt
>
> Normally msgctl08 will complete in a second or two. With this patch I
> don't know how long it will take to complete, and the machine is horridly
> bogged down. It does recover if you manage to kill msgctl08. Feels like
> a terrible memory shortage, but there's plenty of memory free and it isn't
> swapping.
>
>
>

Before the patchset, msgctl08 used to be run with the old msgmni value:

16. Now it is run with a much higher msgmni value (1746 in my case),
since it scales to the memory size.
When I call "msgctl08 100000 16" it completes fast.

Doing the follwing on the ref kernel:
echo 1746 > /proc/sys/kernel/msgmni
msgctl08 100000 1746

makes th test block too :-(

Will check to see where the problem comes from.

Rgards,
Nadia

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

---

Subject: Re: [PATCH 1/8] Scaling msgmni to the amount of lowmem
Posted by Nadia Derbey on Mon, 18 Feb 2008 13:08:42 GMT
View Forum Message <> Reply to Message

Nadia Derbey wrote:
> Andrew Morton wrote:
>
>> On Mon, 11 Feb 2008 15:16:47 +0100 Nadia.Derbey@bull.net wrote:
>>
>>
>>> [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).
>>>
>>
>>
>> Something's wrong here. Running LTP's msgctl08 (specifically:
>> ltp-full-20070228) cripples the machine. It's a 4-way 4GB x86_64.
>>

>> http://userweb.kernel.org/~akpm/config-x.txt
>> http://userweb.kernel.org/~akpm/dmesg-x.txt
>>
>> Normally msgctl08 will complete in a second or two. With this patch I
>> don't know how long it will take to complete, and the machine is horridly
>> bogged down. It does recover if you manage to kill msgctl08. Feels like
>> a terrible memory shortage, but there's plenty of memory free and it
>> isn't
>> swapping.
>>
>>
>>
>
> Before the patchset, msgctl08 used to be run with the old msgmni value:
> 16. Now it is run with a much higher msgmni value (1746 in my case),
> since it scales to the memory size.
> When I call "msgctl08 100000 16" it completes fast.
>
> Doing the follwing on the ref kernel:
> echo 1746 > /proc/sys/kernel/msgmni
> msgctl08 100000 1746
>
> makes th test block too :-(
>
> Will check to see where the problem comes from.
>

Well, actually, the test does not block, it only takes much much more
time to be executed:

doing this:
date; ./msgctl08 100000 XXX; date

gives us the following results:

XXX	16	32	64	128	256	512	1024	1746
time(secs)	2	4	8	16	32	64	132	241

XXX is the # of msg queues to be created = # of processes to be forked
as readers = # of processes to be created as writers
time is approximative since it is obtained by a "date" before and after.

XXX used to be 16 before the patchset ---> 1st column
    --> 16 processes forked as reader
    --> + 16 processes forked as writers
    --> + 16 msg queues
XXX = 1746 (on my victim) after the patchset ---> last column
    --> 1746 reader processes forked

--> + 1746 writers forked
--> + 1746 msg queues created

The same tests on the ref kernel give approximatly the same results.

So if we don't want this longer time to appear as a regression, the LTP
should be changed:
1) either by setting the result of get_max_msgqueues() as the MSGMNI
constant (16) (that would be the best solution in my mind)
2) or by warning the tester that it may take a long time to finish.

There would be 3 tests impacted:

kernel/syscalls/ipc/msgctl/msgctl08.c
kernel/syscalls/ipc/msgctl/msgctl09.c
kernel/syscalls/ipc/msgget/msgget03.c

Cc-ing ltp mailing list ...

Regards,
Nadia

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

---

Subject: Re: [LTP] [PATCH 1/8] Scaling msgmni to the amount of lowmem
Posted by Subrata Modak on Tue, 19 Feb 2008 08:50:55 GMT
View Forum Message <> Reply to Message

> Nadia Derbey wrote:
> > Andrew Morton wrote:
> >
> >> On Mon, 11 Feb 2008 15:16:47 +0100 Nadia.Derbey@bull.net wrote:
> >>
> >>
> >>> [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).
> >>>
> >>
> >>
> >> Something's wrong here. Running LTP's msgctl08 (specifically:
> >> ltp-full-20070228) cripples the machine. It's a 4-way 4GB x86_64.
> >>
> >> http://userweb.kernel.org/~akpm/config-x.txt
> >> http://userweb.kernel.org/~akpm/dmesg-x.txt
> >>
> >> Normally msgctl08 will complete in a second or two. With this patch I
> >> don't know how long it will take to complete, and the machine is horridly
> >> bogged down. It does recover if you manage to kill msgctl08. Feels like
> >> a terrible memory shortage, but there's plenty of memory free and it
> >> isn't
> >> swapping.
> >>
> >>
> >>
> > Before the patchset, msgctl08 used to be run with the old msgmni value:
> > 16. Now it is run with a much higher msgmni value (1746 in my case),
> > since it scales to the memory size.
> > When I call "msgctl08 100000 16" it completes fast.
> >
> > Doing the follwing on the ref kernel:
> > echo 1746 > /proc/sys/kernel/msgmni
> > msgctl08 100000 1746
> >
> > makes th test block too :-(
> >
> > Will check to see where the problem comes from.
> >
>
> Well, actually, the test does not block, it only takes much much more
> time to be executed:
>
> doing this:
> date; ./msgctl08 100000 XXX; date
>
>
> gives us the following results:
> XXX   16   32   64   128   256   512   1024   1746
> time(secs)   2   4   8   16   32   64   132   241
>
> XXX is the # of msg queues to be created = # of processes to be forked

> as readers = # of processes to be created as writers
> time is approximative since it is obtained by a "date" before and after.
>
> XXX used to be 16 before the patchset ---> 1st column
> --> 16 processes forked as reader
> --> + 16 processes forked as writers
> --> + 16 msg queues
> XXX = 1746 (on my victim) after the patchset ---> last column
> --> 1746 reader processes forked
> --> + 1746 writers forked
> --> + 1746 msg queues created
>
> The same tests on the ref kernel give approximatly the same results.
>
> So if we don't want this longer time to appear as a regression, the LTP
> should be changed:
> 1) either by setting the result of get_max_msgqueues() as the MSGMNI
> constant (16) (that would be the best solution in my mind)
> 2) or by warning the tester that it may take a long time to finish.
>
> There would be 3 tests impacted:
>
> kernel/syscalls/ipc/msgctl/msgctl08.c
> kernel/syscalls/ipc/msgctl/msgctl09.c
> kernel/syscalls/ipc/msgget/msgget03.c

We will change the test case if need that be. Nadia, kindly send us the
patch set which will do the necessary changes.

Regards--
Subrata

>
> Cc-ing ltp mailing list ...
>
> Regards,
> Nadia
>
>
>
> ---
> This SF.net email is sponsored by: Microsoft
> Defy all challenges. Microsoft(R) Visual Studio 2008.
> http://clk.atdmt.com/MRT/go/vse0120000070mrt/direct/01/
> _____
> Ltp-list mailing list
> Ltp-list@lists.sourceforge.net
> https://lists.sourceforge.net/lists/listinfo/ltp-list

Subject: Re: [LTP] [PATCH 1/8] Scaling msgmni to the amount of lowmem
Posted by Nadia Derbey on Tue, 19 Feb 2008 17:16:12 GMT
View Forum Message <> Reply to Message

Subrata Modak wrote:
>>Nadia Derbey wrote:
>>
>>>Andrew Morton wrote:
>>>
>>>
>>>>On Mon, 11 Feb 2008 15:16:47 +0100 Nadia.Derbey@bull.net wrote:
>>>>
>>>>
>>>>
>>>>>[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).
>>>>>
>>>>
>>>>
>>>>Something's wrong here. Running LTP's msgctl08 (specifically:
>>>>ltp-full-20070228) cripples the machine. It's a 4-way 4GB x86_64.
>>>>
>>>>http://userweb.kernel.org/~akpm/config-x.txt
>>>>http://userweb.kernel.org/~akpm/dmesg-x.txt
>>>>
>>>>Normally msgctl08 will complete in a second or two. With this patch I
>>>>don't know how long it will take to complete, and the machine is horridly
>>>>bogged down. It does recover if you manage to kill msgctl08. Feels like
>>>>a terrible memory shortage, but there's plenty of memory free and it
>>>>isn't
>>>>swapping.

>>>>
>>>>
>>>>
>>>
>>>Before the patchset, msgctl08 used to be run with the old msgmni value:
>>>16. Now it is run with a much higher msgmni value (1746 in my case),
>>>since it scales to the memory size.
>>>When I call "msgctl08 100000 16" it completes fast.
>>>
>>>Doing the follwing on the ref kernel:
>>>echo 1746 > /proc/sys/kernel/msgmni
>>>msgctl08 100000 1746
>>>
>>>makes th test block too :-(
>>>
>>>Will check to see where the problem comes from.
>>>
>>
>>Well, actually, the test does not block, it only takes much much more
>>time to be executed:
>>
>>doing this:
>>date; ./msgctl08 100000 XXX; date
>>
>>
>>gives us the following results:
>>XXX   16   32   64   128   256   512   1024   1746
>>time(secs)   2   4   8   16   32   64   132   241
>>
>>XXX is the # of msg queues to be created = # of processes to be forked
>>as readers = # of processes to be created as writers
>>time is approximative since it is obtained by a "date" before and after.
>>
>>XXX used to be 16 before the patchset ---> 1st column
>>   --> 16 processes forked as reader
>>   --> + 16 processes forked as writers
>>   --> + 16 msg queues
>>XXX = 1746 (on my victim) after the patchset ---> last column
>>   --> 1746 reader processes forked
>>   --> + 1746 writers forked
>>   --> + 1746 msg queues created
>>
>>The same tests on the ref kernel give approximatly the same results.
>>
>>So if we don't want this longer time to appear as a regression, the LTP
>>should be changed:
>>1) either by setting the result of get_max_msgqueues() as the MSGMNI
>>constant (16) (that would be the best solution in my mind)

>>2) or by warning the tester that it may take a long time to finish.
>>
>>There would be 3 tests impacted:
>>
>>kernel/syscalls/ipc/msgctl/msgctl08.c
>>kernel/syscalls/ipc/msgctl/msgctl09.c
>>kernel/syscalls/ipc/msgget/msgget03.c
>
>
> We will change the test case if need that be. Nadia, kindly send us the
> patch set which will do the necessary changes.
>
> Regards--
> Subrata
>

Subrata,

You'll find the patch in attachment.
FYI I didn't change msgget03.c since we need to get the actual max value
in order to generate an error.

Regards,
Nadia

Subject: Re: [LTP] [PATCH 1/8] Scaling msgmni to the amount of lowmem
Posted by Matt Helsley on Tue, 19 Feb 2008 22:16:58 GMT
View Forum Message <> Reply to Message

On Tue, 2008-02-19 at 18:16 +0100, Nadia Derbey wrote:

<snip>

> +#define MAX_MSGQUEUES 16   /* MSGMNI as defined in linux/msg.h */
> +

It's not quite the maximum anymore, is it? More like the minumum
maximum ;). A better name might better document what the test is
actually trying to do.

One question I have is whether the unpatched test is still valuable.

Based on my limited knowledge of the test I suspect it's still a correct
test of message queues. If so, perhaps renaming the old test (so it's
not confused with a performance regression) and adding your patched
version is best?

<snip>

Cheers,
 -Matt Helsley

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

---

Subject: Re: [LTP] [PATCH 1/8] Scaling msgmni to the amount of lowmem
Posted by Subrata Modak on Wed, 20 Feb 2008 09:44:55 GMT
View Forum Message <> Reply to Message

> Subrata Modak wrote:
> >>Nadia Derbey wrote:
> >>
> >>>Andrew Morton wrote:
> >>>
> >>>
> >>>>On Mon, 11 Feb 2008 15:16:47 +0100 Nadia.Derbey@bull.net wrote:
> >>>>
> >>>>
> >>>>
> >>>>>[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).
> >>>>>
> >>>>>
> >>>>>
> >>>>Something's wrong here. Running LTP's msgctl08 (specifically:
> >>>>ltp-full-20070228) cripples the machine. It's a 4-way 4GB x86_64.
> >>>>

> >>>>http://userweb.kernel.org/~akpm/config-x.txt
> >>>>http://userweb.kernel.org/~akpm/dmesg-x.txt
> >>>>
> >>>>Normally msgctl08 will complete in a second or two. With this patch I
> >>>>don't know how long it will take to complete, and the machine is horridly
> >>>>bogged down. It does recover if you manage to kill msgctl08. Feels like
> >>>>a terrible memory shortage, but there's plenty of memory free and it
> >>>>isn't
> >>>>swapping.
> >>>>
> >>>>
> >>>>
> >>>
> >>>Before the patchset, msgctl08 used to be run with the old msgmni value:
> >>>16. Now it is run with a much higher msgmni value (1746 in my case),
> >>>since it scales to the memory size.
> >>>When I call "msgctl08 100000 16" it completes fast.
> >>>
> >>>Doing the follwing on the ref kernel:
> >>>echo 1746 > /proc/sys/kernel/msgmni
> >>>msgctl08 100000 1746
> >>>
> >>>makes th test block too :-(
> >>>
> >>>Will check to see where the problem comes from.
> >>>
> >>
> >>Well, actually, the test does not block, it only takes much much more
> >>time to be executed:
> >>
> >>doing this:
> >>date; ./msgctl08 100000 XXX; date
> >>
> >>
> >>gives us the following results:
> >>XXX   16   32   64   128   256   512   1024   1746
> >>time(secs)   2   4   8   16   32   64   132   241
> >>
> >>XXX is the # of msg queues to be created = # of processes to be forked
> >>as readers = # of processes to be created as writers
> >>time is approximative since it is obtained by a "date" before and after.
> >>
> >>XXX used to be 16 before the patchset ---> 1st column
> >>   --> 16 processes forked as reader
> >>   --> + 16 processes forked as writers
> >>   --> + 16 msg queues
> >>XXX = 1746 (on my victim) after the patchset ---> last column
> >>   --> 1746 reader processes forked

> >>   --> + 1746 writers forked
> >>   --> + 1746 msg queues created
> >>
> >>The same tests on the ref kernel give approximatly the same results.
> >>
> >>So if we don't want this longer time to appear as a regression, the LTP
> >>should be changed:
> >>1) either by setting the result of get_max_msgqueues() as the MSGMNI
> >>constant (16) (that would be the best solution in my mind)
> >>2) or by warning the tester that it may take a long time to finish.
> >>
> >>There would be 3 tests impacted:
> >>
> >>kernel/syscalls/ipc/msgctl/msgctl08.c
> >>kernel/syscalls/ipc/msgctl/msgctl09.c
> >>kernel/syscalls/ipc/msgget/msgget03.c
> >
> >
> > We will change the test case if need that be. Nadia, kindly send us the
> > patch set which will do the necessary changes.
> >
> > Regards--
> > Subrata
> >
>
> Subrata,
>
> You'll find the patch in attachment.
> FYI I didn't change msgget03.c since we need to get the actual max value
> in order to generate an error.

Thanks. The same has been Merged.

Regards--
Subrata

>
> Regards,
> Nadia
>

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

Subject: Re: [LTP] [PATCH 1/8] Scaling msgmni to the amount of lowmem
Posted by Nadia Derbey on Thu, 21 Feb 2008 08:39:07 GMT
View Forum Message <> Reply to Message

Matt Helsley wrote:
> On Tue, 2008-02-19 at 18:16 +0100, Nadia Derbey wrote:
>
> <snip>
>
>>+#define MAX_MSGQUEUES 16   /* MSGMNI as defined in linux/msg.h */
>>+
>
>
> It's not quite the maximum anymore, is it? More like the minumum
> maximum ;). A better name might better document what the test is
> actually trying to do.
>
> One question I have is whether the unpatched test is still valuable.
> Based on my limited knowledge of the test I suspect it's still a correct
> test of message queues. If so, perhaps renaming the old test (so it's
> not confused with a performance regression) and adding your patched
> version is best?
>

Yes, you're completely right.

I'll resend a patch today.

Regards,
Nadia

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

Subject: Re: [LTP] [PATCH 1/8] Scaling msgmni to the amount of lowmem
Posted by Nadia Derbey on Thu, 21 Feb 2008 12:36:42 GMT
View Forum Message <> Reply to Message

Matt Helsley wrote:
> On Tue, 2008-02-19 at 18:16 +0100, Nadia Derbey wrote:
>
> <snip>
>
>>+#define MAX_MSGQUEUES 16   /* MSGMNI as defined in linux/msg.h */
>>+
>

>
> It's not quite the maximum anymore, is it? More like the minumum
> maximum ;). A better name might better document what the test is
> actually trying to do.
>
> One question I have is whether the unpatched test is still valuable.
> Based on my limited knowledge of the test I suspect it's still a correct
> test of message queues. If so, perhaps renaming the old test (so it's
> not confused with a performance regression) and adding your patched
> version is best?
>

So, here's the new patch based on Matt's points.

Subrata, it has to be applied on top of the original ltp-full-20080131.
Please tell me if you'd prefer one based on the merged version you've
got (i.e. with my Tuesday patch applied).

Regards,
Nadia

Subject: Re: [LTP] [PATCH 1/8] Scaling msgmni to the amount of lowmem
Posted by Nadia Derbey on Thu, 21 Feb 2008 13:02:00 GMT
View Forum Message <> Reply to Message

Nadia Derbey wrote:
> Matt Helsley wrote:
>
>> On Tue, 2008-02-19 at 18:16 +0100, Nadia Derbey wrote:
>>
>> <snip>
>>
>>> +#define MAX_MSGQUEUES 16   /* MSGMNI as defined in linux/msg.h */
>>> +
>>
>>
>>
>> It's not quite the maximum anymore, is it? More like the minumum
>> maximum ;). A better name might better document what the test is
>> actually trying to do.
>>
>> One question I have is whether the unpatched test is still valuable.
>> Based on my limited knowledge of the test I suspect it's still a correct

>> test of message queues. If so, perhaps renaming the old test (so it's
>> not confused with a performance regression) and adding your patched
>> version is best?
>>
>
> So, here's the new patch based on Matt's points.
>
> Subrata, it has to be applied on top of the original ltp-full-20080131.
> Please tell me if you'd prefer one based on the merged version you've
> got (i.e. with my Tuesday patch applied).
>

Forgot the patch, sorry for that (thx Andrew).

Regards,
Nadia

Subject: Re: [LTP] [PATCH 1/8] Scaling msgmni to the amount of lowmem
Posted by Subrata Modak on Thu, 21 Feb 2008 13:39:38 GMT
View Forum Message <> Reply to Message

> Nadia Derbey wrote:
> > Matt Helsley wrote:
> >
> >> On Tue, 2008-02-19 at 18:16 +0100, Nadia Derbey wrote:
> >>
> >> <snip>
> >>
> >>> +#define MAX_MSGQUEUES 16   /* MSGMNI as defined in linux/msg.h */
> >>> +
> >>
> >>
> >>
> >> It's not quite the maximum anymore, is it? More like the minumum
> >> maximum ;). A better name might better document what the test is
> >> actually trying to do.
> >>
> >> One question I have is whether the unpatched test is still valuable.
> >> Based on my limited knowledge of the test I suspect it's still a correct
> >> test of message queues. If so, perhaps renaming the old test (so it's
> >> not confused with a performance regression) and adding your patched

> >> version is best?
> >>
> >
> > So, here's the new patch based on Matt's points.
> >
> > Subrata, it has to be applied on top of the original ltp-full-20080131.
> > Please tell me if you'd prefer one based on the merged version you've
> > got (i.e. with my Tuesday patch applied).

Nadia, I would prefer Patch on the top of the already merged version (on top of latest CVS snapshot as of today). Anyways, thanks for all these effort :-)

--Subrata

> >
>
> Forgot the patch, sorry for that (thx Andrew).
>
> Regards,
> Nadia
>

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

---

Subject: Re: [LTP] [PATCH 1/8] Scaling msgmni to the amount of lowmem
Posted by Nadia Derbey on Fri, 22 Feb 2008 06:25:20 GMT
View Forum Message <> Reply to Message

Subrata Modak wrote:
>>Nadia Derbey wrote:
>>
>>>Matt Helsley wrote:
>>>
>>>
>>>>On Tue, 2008-02-19 at 18:16 +0100, Nadia Derbey wrote:
>>>>
>>>>><snip>
>>>>
>>>>+#define MAX_MSGQUEUES 16   /* MSGMNI as defined in linux/msg.h */
>>>>+
>>>>
>>>>
>>>>

>>>>It's not quite the maximum anymore, is it? More like the minumum
>>>>maximum ;). A better name might better document what the test is
>>>>actually trying to do.
>>>>
>>>>One question I have is whether the unpatched test is still valuable.
>>>>Based on my limited knowledge of the test I suspect it's still a correct
>>>>test of message queues. If so, perhaps renaming the old test (so it's
>>>>not confused with a performance regression) and adding your patched
>>>>version is best?
>>>>
>>>
>>>So, here's the new patch based on Matt's points.
>>>
>>>Subrata, it has to be applied on top of the original ltp-full-20080131.
>>>Please tell me if you'd prefer one based on the merged version you've
>>>got (i.e. with my Tuesday patch applied).
>
>
> Nadia, I would prefer Patch on the top of the already merged version (on
> top of latest CVS snapshot as of today). Anyways, thanks for all these
> effort :-)
>
> --Subrata
>

In attachment, you'll find a patch to apply on top of the patches I sent
you on Tuesday.

Regards,
Nadia

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

---

Subject: Re: [LTP] [PATCH 1/8] Scaling msgmni to the amount of lowmem
Posted by Subrata Modak on Fri, 22 Feb 2008 08:41:26 GMT
View Forum Message <> Reply to Message

On Fri, 2008-02-22 at 07:25 +0100, Nadia Derbey wrote:
> Subrata Modak wrote:
> >>Nadia Derbey wrote:
> >>
> >>>Matt Helsley wrote:
> >>>
> >>>

> >>>>On Tue, 2008-02-19 at 18:16 +0100, Nadia Derbey wrote:
> >>>>
> >>>><snip>
> >>>>
> >>>>>+#define MAX_MSGQUEUES 16   /* MSGMNI as defined in linux/msg.h */
> >>>>>+
> >>>>
> >>>>
> >>>>
> >>>>It's not quite the maximum anymore, is it? More like the minumum
> >>>>maximum ;). A better name might better document what the test is
> >>>>actually trying to do.
> >>>>
> >>>>One question I have is whether the unpatched test is still valuable.
> >>>>Based on my limited knowledge of the test I suspect it's still a correct
> >>>>test of message queues. If so, perhaps renaming the old test (so it's
> >>>>not confused with a performance regression) and adding your patched
> >>>>version is best?
> >>>>
> >>>
> >>>So, here's the new patch based on Matt's points.
> >>>
> >>>Subrata, it has to be applied on top of the original ltp-full-20080131.
> >>>Please tell me if you'd prefer one based on the merged version you've
> >>>got (i.e. with my Tuesday patch applied).
> >
> >
> > Nadia, I would prefer Patch on the top of the already merged version (on
> > top of latest CVS snapshot as of today). Anyways, thanks for all these
> > effort :-)
> >
> > --Subrata
> >
>
> In attachment, you'll find a patch to apply on top of the patches I sent
> you on Tuesday.

Nadia,

Thanks a ton for that. The same has been merged.

Regards--
Subrata

>
> Regards,
> Nadia

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

Subject: Re: [PATCH 1/8] Scaling msgmni to the amount of lowmem
Posted by tony.luck on Tue, 29 Apr 2008 20:28:14 GMT
View Forum Message <> Reply to Message

On Mon, Feb 11, 2008 at 7:16 AM, <Nadia.Derbey@bull.net> wrote:
> 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
...
> +out_callback:
> +
> + printk(KERN_INFO "msgmni has been set to %d for ipc namespace %p\n",
> + ns->msg_ctlmni, ns);
> +}

This patch has now made its way to mainline. I can see how this printk was really useful to you while developing this patch. But does it add much value in a production system? It just looks like another piece of clutter on the console to my uncontainerized eyes.

-Tony

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

Subject: Re: [PATCH 1/8] Scaling msgmni to the amount of lowmem
Posted by Nadia Derbey on Mon, 05 May 2008 08:45:11 GMT
View Forum Message <> Reply to Message

Tony Luck wrote:
> On Mon, Feb 11, 2008 at 7:16 AM, <Nadia.Derbey@bull.net> wrote:
>
>> 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
>
> ...

>
>> +out_callback:
>> +
>> +	printk(KERN_INFO "msgmni has been set to %d for ipc namespace %p\n",
>> +		ns->msg_ctlmni, ns);
>> +}
>
>
> This patch has now made its way to mainline. I can see how this printk
> was really useful to you while developing this patch. But does it add
> much value in a production system? It just looks like another piece of
> clutter on the console to my uncontainerized eyes.
>
> -Tony
>
>

Well, this printk had been suggested by somebody (sorry I don't remember
who) when I first submitted the patch. Actually I think it might be
useful for a sysadmin to be aware of a change in the msgmni value: we
have the message not only at boot time, but also each time msgmni is
recomputed because of a change in the amount of memory.
Also, at boot time, I think it's interesting to have the actual msgmni
value: it used to unconditionally be set to 16. Some applications that
used to need an initialization script setting msgmni to a higher value
might not need that script anymore, since the new value might fit their
needs.

Regards,
Nadia

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

---

Subject: RE: [PATCH 1/8] Scaling msgmni to the amount of lowmem
Posted by tony.luck on Tue, 06 May 2008 16:42:25 GMT
View Forum Message <> Reply to Message

> Well, this printk had been suggested by somebody (sorry I don't remember
> who) when I first submitted the patch. Actually I think it might be
> useful for a sysadmin to be aware of a change in the msgmni value: we
> have the message not only at boot time, but also each time msgmni is
> recomputed because of a change in the amount of memory.

If the message is directed at the system administrator, then it would

be nice if there were some more meaningful way to show the namespace
that is affected than just printing the hex address of the kernel structure.

As the sysadmin for my test systems, printing the hex address is mildly
annoying ... I now have to add a new case to my scripts that look at
dmesg output for unusual activity.

Is there some better "name for a namespace" than the address? Perhaps
the process id of the process that instantiated the namespace???

-Tony

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

---

Subject: Re: [PATCH 1/8] Scaling msgmni to the amount of lowmem
Posted by serue on Tue, 06 May 2008 18:05:27 GMT
View Forum Message <> Reply to Message

Quoting Luck, Tony (tony.luck@intel.com):
> > Well, this printk had been suggested by somebody (sorry I don't remember
> > who) when I first submitted the patch. Actually I think it might be
> > useful for a sysadmin to be aware of a change in the msgmni value: we
> > have the message not only at boot time, but also each time msgmni is
> > recomputed because of a change in the amount of memory.
>
> If the message is directed at the system administrator, then it would
> be nice if there were some more meaningful way to show the namespace
> that is affected than just printing the hex address of the kernel structure.
>
> As the sysadmin for my test systems, printing the hex address is mildly
> annoying ... I now have to add a new case to my scripts that look at
> dmesg output for unusual activity.
>
> Is there some better "name for a namespace" than the address? Perhaps
> the process id of the process that instantiated the namespace???

I agree with Tony here. Aside from the nuisance it is to see that
message on console every time I unshare a namespace, a printk doesn't
seem like the right way to output the info. At most I'd say an audit
message.

-serge

_____
Containers mailing list
Containers@lists.linux-foundation.org

Subject: Re: [PATCH 1/8] Scaling msgmni to the amount of lowmem
Posted by Nadia Derbey on Wed, 07 May 2008 05:13:10 GMT
View Forum Message <> Reply to Message

Luck, Tony wrote:
>>Well, this printk had been suggested by somebody (sorry I don't remember
>>who) when I first submitted the patch. Actually I think it might be
>>useful for a sysadmin to be aware of a change in the msgmni value: we
>>have the message not only at boot time, but also each time msgmni is
>>recomputed because of a change in the amount of memory.
>
>
> If the message is directed at the system administrator, then it would
> be nice if there were some more meaningful way to show the namespace
> that is affected than just printing the hex address of the kernel structure.
>
> As the sysadmin for my test systems, printing the hex address is mildly
> annoying ... I now have to add a new case to my scripts that look at
> dmesg output for unusual activity.
>
> Is there some better "name for a namespace" than the address? Perhaps
> the process id of the process that instantiated the namespace???
>

Unfortunately no when we are inside an ipc namespace, we don't have such
interesting informations. But I agree with you, an address is not
readable enough. I'll try to find a solution.

Regards,
Nadia

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

Subject: Re: [PATCH 1/8] Scaling msgmni to the amount of lowmem
Posted by Nadia Derbey on Wed, 07 May 2008 05:37:11 GMT
View Forum Message <> Reply to Message

Serge E. Hallyn wrote:
> Quoting Luck, Tony (tony.luck@intel.com):
>

>>>Well, this printk had been suggested by somebody (sorry I don't remember
>>>who) when I first submitted the patch. Actually I think it might be
>>>useful for a sysadmin to be aware of a change in the msgmni value: we
>>>have the message not only at boot time, but also each time msgmni is
>>>recomputed because of a change in the amount of memory.
>>
>>If the message is directed at the system administrator, then it would
>>be nice if there were some more meaningful way to show the namespace
>>that is affected than just printing the hex address of the kernel structure.
>>
>>As the sysadmin for my test systems, printing the hex address is mildly
>>annoying ... I now have to add a new case to my scripts that look at
>>dmesg output for unusual activity.
>>
>>Is there some better "name for a namespace" than the address? Perhaps
>>the process id of the process that instantiated the namespace???
>
>
> I agree with Tony here. Aside from the nuisance it is to see that
> message on console every time I unshare a namespace, a printk doesn't
> seem like the right way to output the info.

But you agree that this is happening only because you're doing tests
related to namespaces, right?
I don't think that in a "standard" configuration this will happen very
frequently, but may be I'm wrong.

> At most I'd say an audit
> message.
>

That's a good idea. Thanks, Serge. I'll do that.

Regards,
Nadia

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

---

Subject: Re: [PATCH 1/8] Scaling msgmni to the amount of lowmem
Posted by serue on Wed, 07 May 2008 13:17:12 GMT
View Forum Message <> Reply to Message

Quoting Nadia Derbey (Nadia.Derbey@bull.net):
> Serge E. Hallyn wrote:

>> Quoting Luck, Tony (tony.luck@intel.com):
>>>> Well, this printk had been suggested by somebody (sorry I don't remember
>>>> who) when I first submitted the patch. Actually I think it might be
>>>> useful for a sysadmin to be aware of a change in the msgmni value: we
>>>> have the message not only at boot time, but also each time msgmni is
>>>> recomputed because of a change in the amount of memory.
>>>
>>> If the message is directed at the system administrator, then it would
>>> be nice if there were some more meaningful way to show the namespace
>>> that is affected than just printing the hex address of the kernel
>>> structure.
>>>
>>> As the sysadmin for my test systems, printing the hex address is mildly
>>> annoying ... I now have to add a new case to my scripts that look at
>>> dmesg output for unusual activity.
>>>
>>> Is there some better "name for a namespace" than the address? Perhaps
>>> the process id of the process that instantiated the namespace???
>> I agree with Tony here. Aside from the nuisance it is to see that
>> message on console every time I unshare a namespace, a printk doesn't
>> seem like the right way to output the info.
>
> But you agree that this is happening only because you're doing tests
> related to namespaces, right?

Yup :)

> I don't think that in a "standard" configuration this will happen very
> frequently, but may be I'm wrong.
>
>> At most I'd say an audit
>> message.
>
> That's a good idea. Thanks, Serge. I'll do that.

It'll probably still end up a printk for me, but it'll be my own fault
for not setting up audit.

> Regards,
> Nadia

thanks,
-serge

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

Subject: Re: [PATCH 1/8] Scaling msgmni to the amount of lowmem
Posted by Matt Helsley on Wed, 07 May 2008 18:12:07 GMT
View Forum Message <> Reply to Message

On Wed, 2008-05-07 at 08:17 -0500, Serge E. Hallyn wrote:
> Quoting Nadia Derbey (Nadia.Derbey@bull.net):
> > Serge E. Hallyn wrote:
> >> Quoting Luck, Tony (tony.luck@intel.com):
> >>>> Well, this printk had been suggested by somebody (sorry I don't remember
> >>>> who) when I first submitted the patch. Actually I think it might be
> >>>> useful for a sysadmin to be aware of a change in the msgmni value: we
> >>>> have the message not only at boot time, but also each time msgmni is
> >>>> recomputed because of a change in the amount of memory.
> >>>
> >>> If the message is directed at the system administrator, then it would
> >>> be nice if there were some more meaningful way to show the namespace
> >>> that is affected than just printing the hex address of the kernel
> >>> structure.
> >>>
> >>> As the sysadmin for my test systems, printing the hex address is mildly
> >>> annoying ... I now have to add a new case to my scripts that look at
> >>> dmesg output for unusual activity.
> >>>
> >>> Is there some better "name for a namespace" than the address? Perhaps
> >>> the process id of the process that instantiated the namespace???
> >> I agree with Tony here. Aside from the nuisance it is to see that
> >> message on console every time I unshare a namespace, a printk doesn't
> >> seem like the right way to output the info.
> >
> > But you agree that this is happening only because you're doing tests
> > related to namespaces, right?
>
> Yup :)
>
> > I don't think that in a "standard" configuration this will happen very
> > frequently, but may be I'm wrong.
> >
> >> At most I'd say an audit
> >> message.
>
> > That's a good idea. Thanks, Serge. I'll do that.

 I'm not familiar with kernel policies regarding audit messages. Are
audit messages treated anything like kernel interfaces when it comes to
removing/changing them?

Cheers,
 -Matt Helsley

_____

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