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