Subject: [PATCH 0/8] Change default MSGMNI tunable to scale with lowmem (v3)
Posted by Nadia Derbey on Mon, 11 Feb 2008 14:16:46 GMT
View Forum Message <> Reply to Message

Resending the set of patches after Yasunori's remark about being able to
turn on/off automatic recomputing.
(see message at http://lkml.org/lkml/2008/2/5/149).
I actually introduced an intermediate solution: when msgmni is set by hand,
it is uneregistered from the ipcns notifier chain (i.e. automatic recomputing
is disabled). This corresponds to an implicit turn off. Setting it to a
negative value makes it registered back in the notifier chain (which
corresponds to the turn on proposed by Yasunaori).

Only patch # 8 introduces new stuff compared to the already sent patch sets.

Also ported the patchset to 2.6.24-mm1.

On large systems we'd like to allow a larger number of message queues.
In some cases up to 32K. However simply setting MSGMNI to a larger value may
cause problems for smaller systems.

The first patch of this series introduces a default maximum number of message
queue ids that scales with the amount of lowmem.

Since msgmni is per namespace and there is no amount of memory dedicated to
each namespace so far, the second patch of this series scales msgmni to
the number of ipc namespaces too.

Since msgmni depends on the amount of memory, it becomes necessary to
recompute it upon memory add/remove.
In the 4th patch, memory hotplug management is added: a notifier block is
registered into the memory hotplug notifier chain for the ipc subsystem.
Since the ipc namespaces are not linked together, they have their own
notification chain: one notifier_block is defined per ipc namespace.
Each time an ipc namespace is created (removed) it registers (unregisters)
its notifier block in (from) the ipcns chain.
The callback routine registered in the memory chain invokes the ipcns
notifier chain with the IPCNS_MEMCHANGE event.
Each callback routine registered in the ipcns namespace, in turn, recomputes
msgmni for the owning namespace.

The 5th patch makes it possible to keep the memory hotplug notifier chain's
lock for a lesser amount of time: instead of directly notifying the ipcns
notifier chain upon memory add/remove, a work item is added to the global
workqueue. When activated, this work item is the one who notifies the ipcns
notifier chain.

Since msgmni depends on the number of ipc namespaces, it becomes necessary
to recompute it upon ipc namespace creation / removal.
The 6th patch uses the ipc namespace notifier chain for that purpose: that
chain is notified each time an ipc namespace is created or removed. This makes
it possible to recompute msgmni for all the namespaces each time one of them
is created or removed.

When msgmni is explicitely set from userspace, we should avoid recomputing it
upon memory add/remove or ipcns creation/removal.
This is what the 7th patch does: it simply unregisters the ipcns callback
routine as soon as msgmni has been changed from procfs or sysctl().

Even if msgmni is set by hand, it should be possible to make it back
automatically recomputed upon memory add/remove or ipcns creation/removal.
This what is achieved in patch 8: if set to a negative value, msgmni is added
back to the ipcns notifier chain, making it automatically recomputed again.

These patches should be applied to 2.6.24-mm1, in the following order:

[PATCH 1/8]: ipc_scale_msgmni_with_lowmem.patch
[PATCH 2/8]: ipc_scale_msgmni_with_namespaces.patch
[PATCH 3/8]: ipc_slab_memory_callback_prio_to_const.patch
[PATCH 4/8]: ipc_recompute_msgmni_on_memory_hotplug.patch
[PATCH 5/8]: ipc_ipcns_notification_workqueue.patch
[PATCH 6/8]: ipc_recompute_msgmni_on_ipcns_create_remove.patch
[PATCH 7/8]: ipc_unregister_callback_on_msgmni_manual_change.patch
[PATCH 8/8]: ipc_register_callback_on_negative_msgmni.patch

Andrew, do you think this patchset could be considered for inclusion in -mm
(or at least patches 1 to 6)?

Regards,
Nadia

--

_____