
Subject: Re: [PATCH 01/29] task containersv11 basic task container framework
Posted by [Paul Menage](#) on Sun, 30 Sep 2007 07:10:18 GMT
[View Forum Message](#) <> [Reply to Message](#)

On 9/29/07, Paul Jackson <pj@sgi.com> wrote:

> Paul M:

>

> This patch doesn't build for me in the following case. If I apply the
> rest of the containersv11 patches, it builds, but if I happen to bisect
> into this set of patches having applied only:

These patches weren't the normal style of incremental patchset - they were direct substitutions of the previous patches in Andrew's -mm tree.

Andrew's approach when encountering patches that are, e.g. missing includes on some architectures, is to add a *-fix.patch to the series; in this case it looks as though the fix for the missing ia64 include was added towards the end of the series rather than directly after the patch that needed it. Hence the fact that you couldn't compile until that later patch had been pushed.

Andrew, is it practical for you to collapse any *-fix.patch files for cgroups into the appropriate patches that they fix? That would simplify the patch set a bit.

>

> task-containersv11-basic-task-container-framework.patch

>

> while using sn2_defconfig (with CONFIG_CGROUPS=y), then a build for
> arch ia64 fails with:

>

>

> \$ make kernel/cgroup.o

> CHK include/linux/version.h

> CHK include/linux/utsrelease.h

> CALL scripts/checksyscalls.sh

> <stdin>:1389:2: warning: #warning syscall revokeat not implemented

> <stdin>:1393:2: warning: #warning syscall frevoke not implemented

> CC kernel/cgroup.o

> kernel/cgroup.c: In function 'cgroup_new_inode':

> kernel/cgroup.c:227: error: variable 'cgroup_backing_dev_info' has initializer but incomplete type

> kernel/cgroup.c:228: error: unknown field 'capabilities' specified in initializer

> kernel/cgroup.c:228: error: 'BDI_CAP_NO_ACCT_DIRTY' undeclared (first use in this function)

> kernel/cgroup.c:228: error: (Each undeclared identifier is reported only once

> kernel/cgroup.c:228: error: for each function it appears in.)

> kernel/cgroup.c:228: error: 'BDI_CAP_NO_WRITEBACK' undeclared (first use in this function)

> kernel/cgroup.c:228: warning: excess elements in struct initializer

> kernel/cgroup.c:228: warning: (near initialization for 'cgroup_backing_dev_info')
> kernel/cgroup.c:227: error: storage size of 'cgroup_backing_dev_info' isn't known
> kernel/cgroup.c:227: warning: unused variable 'cgroup_backing_dev_info'
> make[1]: *** [kernel/cgroup.o] Error 1
> make: *** [kernel/cgroup.o] Error 2
>
>
> I haven't tested building other partial applications of the
> containersv11 patch set; any initial sequence of them should build.

This wasn't strictly the containersv11 patch set - it was a set of patches from -mm that needed to be replaced in order to rename containers to control groups. The initial subset of these patches was the same as containersv11, but it also included a bunch of other people's patches (containerstats, memory controller, etc).

>
> ... also ... too bad the names of these patches all have 'container' in
> them, not 'cgroup'. I realize how this came to be, due to changing
> container to cgroup after these patches were named, but it sure would
> be nice if the final patch set record that gets layed down in Linus's
> history showed the correct name of 'cgroup' in these patch names.

Agreed. Presumably that would be fairly straightforward for Andrew to do directly on his quilt tree - I kept the names the same as the previous patches to make it simpler to match up old to new.

Paul

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