Subject: Re: [PATCH 1/2] rcfs core patch
Posted by ebiederm on Thu, 08 Mar 2007 03:12:00 GMT
View Forum Message <> Reply to Message

Srivatsa Vaddagiri <vatsa@in.ibm.com> writes:

> Heavily based on Paul Menage's (inturn cpuset) work. The big difference
> is that the patch uses task->nsproxy to group tasks for resource control
> purpose (instead of task->containers).
>
> The patch retains the same user interface as Paul Menage's patches. In
> particular, you can have multiple hierarchies, each hierarchy giving a
> different composition/view of task-groups.
>
> (Ideally this patch should have been split into 2 or 3 sub-patches, but
> will do that on a subsequent version post)

After looking at the discussion that happened immediately after this was
posted this feels like the right general direction to get the different
parties talking to each other. I'm not convinced about the whole idea
yet but this looks like a step in a useful direction.

I have a big request.

Please next time this kind of patch is posted add a description of
what is happening and why. I have yet to see people explain why
this is a good idea. Why the current semantics were chosen.

The review is still largely happening at the why level but no
one is addressing that yet. So please can we have a why.

I have a question? What does rcfs look like if we start with
the code that is in the kernel? That is start with namespaces
and nsproxy and just build a filesystem to display/manipulate them?
With the code built so it will support adding resource controllers
when they are ready?

> Signed-off-by : Srivatsa Vaddagiri <vatsa@in.ibm.com>
> Signed-off-by : Paul Menage <menage@google.com>
>
>
> ---
>
> linux-2.6.20-vatsa/include/linux/init_task.h | 4
> linux-2.6.20-vatsa/include/linux/nsproxy.h | 5
> linux-2.6.20-vatsa/init/Kconfig | 22
> linux-2.6.20-vatsa/init/main.c | 1
> linux-2.6.20-vatsa/kernel/Makefile | 1

>
>
> ---
>

> diff -puN include/linux/nsproxy.h~rcfs include/linux/nsproxy.h
> --- linux-2.6.20/include/linux/nsproxy.h~rcfs 2007-03-01 14:20:47.000000000
> +0530
> +++ linux-2.6.20-vatsa/include/linux/nsproxy.h 2007-03-01 14:20:47.000000000
> +0530
> @@ -28,6 +28,10 @@ struct nsproxy {
We probably want to rename this struct task_proxy....
And then we can rename most of the users things like:
dup_task_proxy, clone_task_proxy, get_task_proxy, free_task_proxy,
put_task_proxy, exit_task_proxy, init_task_proxy....

>   struct ipc_namespace *ipc_ns;
>   struct mnt_namespace *mnt_ns;
>   struct pid_namespace *pid_ns;
> +#ifdef CONFIG_RCFS
> + struct list_head list;

This extra list of nsproxy's is unneeded and a performance problem the
way it is used. In general we want to talk about the individual resource
controllers not the nsproxy.

> + void *ctlr_data[CONFIG_MAX_RC_SUBSYS];

I still don't understand why these pointers are so abstract,
and why we need an array lookup into them?

> +#endif
> };
> extern struct nsproxy init_nsproxy;
>
> @@ -35,6 +39,12 @@ struct nsproxy *dup_namespaces(struct ns
> int copy_namespaces(int flags, struct task_struct *tsk);
> void get_task_namespaces(struct task_struct *tsk);
> void free_nsproxy(struct nsproxy *ns);
> +#ifdef CONFIG_RCFS
> +struct nsproxy *find_nsproxy(struct nsproxy *ns);
> +int namespaces_init(void);
> +#else
> +static inline int namespaces_init(void) { return 0;}
> +#endif
>
> static inline void put_nsproxy(struct nsproxy *ns)

> {
> diff -puN /dev/null include/linux/rcfs.h
> --- /dev/null 2006-02-25 03:06:56.000000000 +0530
> +++ linux-2.6.20-vatsa/include/linux/rcfs.h 2007-03-01 14:20:47.000000000 +0530
> @@ -0,0 +1,72 @@
> +#ifndef _LINUX_RCFS_H
> +#define _LINUX_RCFS_H
> +
> +#ifdef CONFIG_RCFS
> +
> +/* struct cftype:
> + *
> + * The files in the container filesystem mostly have a very simple read/write
> + * handling, some common function will take care of it. Nevertheless some cases
> + * (read tasks) are special and therefore I define this structure for every
> + * kind of file.

I'm still inclined to think this should be part of /proc, instead of a purely
separate fs. But I might be missing something.

Eric

_____
Containers mailing list
Containers@lists.osdl.org
https://lists.osdl.org/mailman/listinfo/containers