Subject: Re: [PATCH 6/7] containers (V7): BeanCounters over generic process containers
Posted by xemul on Tue, 13 Feb 2007 08:52:24 GMT
View Forum Message <> Reply to Message

menage@google.com wrote:
> This patch implements the BeanCounter resource control abstraction
> over generic process containers. It contains the beancounter core
> code, plus the numfiles resource counter. It doesn't currently contain
> any of the memory tracking code or the code for switching beancounter
> context in interrupts.

Numfiles is not the most interesting place in beancounters.
Kmemsize accounting is much more important actually.

> Currently all the beancounters resource counters are lumped into a
> single hierarchy; ideally it would be possible for each resource
> counter to be a separate container subsystem, allowing them to be
> connected to different hierarchies.
>
> ---
> fs/file_table.c | 11 +
> include/bc/beancounter.h | 192 ++++++++++++++++++++++
> include/bc/misc.h | 27 +++
> include/linux/fs.h | 3
> init/Kconfig | 4
> init/main.c | 3
> kernel/Makefile | 1
> kernel/bc/Kconfig | 17 ++
> kernel/bc/Makefile | 7
> kernel/bc/beancounter.c | 371 ++
> kernel/bc/misc.c | 56 +++++++
> 11 files changed, 691 insertions(+), 1 deletion(-)
>

[snip]

> Index: container-2.6.20/kernel/bc/misc.c
> ===
> --- /dev/null
> +++ container-2.6.20/kernel/bc/misc.c
> @@ -0,0 +1,56 @@
> +
> +#include <linux/fs.h>
> +#include <bc/beancounter.h>
> +
> +int bc_file_charge(struct file *file)
> +{

```
> + int sev;
> + struct beancounter *bc;
> +
> + task_lock(current);
> + bc = task_bc(current);
> + css_get_current(&bc->css);
> + task_unlock(current);
> +
> + sev = (capable(CAP_SYS_ADMIN) ? BC_LIMIT : BC_BARRIER);
> +
> + if (bc_charge(bc, BC_NUMFILES, 1, sev)) {
> +  css_put(&bc->css);
> +  return -EMFILE;
> + }
> +
> + file->f_bc = bc;
> + return 0;
> +}
> +
```

I have already pointed out the fact that this place
will hurt performance too much. If we have some context
on task this context must

1. be get-ed without any locking
2. be settable to some temporary one without
   locking as well

Unfortunately current containers implementation doesn't
allow all of the above which blocks the rest implementation
of beancounters over them.
