
Subject: Re: [RFC][PATCH][1/4] RSS controller setup
Posted by [Paul Menage](#) on Mon, 19 Feb 2007 09:18:55 GMT
[View Forum Message](#) <> [Reply to Message](#)

On 2/19/07, Andrew Morton <akpm@linux-foundation.org> wrote:

>
> This output is hard to parse and to extend. I'd suggest either two
> separate files, or multi-line output:
>
> usage: %lu kB
> limit: %lu kB

Two separate files would be the container usage model that I envisaged, inherited from the way cpusets does things.

And in this case, it should definitely be the limit in one file, readable and writeable, and the usage in another, probably only readable.

Having to read a file called memctlr_usage to find the current limit sounds wrong.

Hmm, I don't appear to have documented this yet, but I think a good naming scheme for container files is <subsystem>.<whatever> - i.e. these should be memctlr.usage and memctlr.limit. The existing grandfathered Cpusets names violate this, but I'm not sure there's a lot we can do about that.

```
> > +static int memctlr_populate(struct container_subsys *ss,
> > +                          struct container *cont)
> > +{
> > +    int rc;
> > +    if ((rc = container_add_file(cont, &memctlr_usage)) < 0)
> > +        return rc;
> > +    if ((rc = container_add_file(cont, &memctlr_limit)) < 0)
> >
> Clean up the first file here?
```

Containers don't currently provide an API for a subsystem to clean up files from a directory - that's done automatically when the directory is deleted.

I think I'll probably change the API for container_add_file to return void, but mark an error in the container itself if something goes wrong - that way rather than all the subsystems having to check for error, container_populate_dir() can do so at the end of calling all the subsystems' populate methods.

Paul
