KAMEZAWA Hiroyuki wrote:
> On Fri, 15 Feb 2008 12:44:18 -0800
> Paul Menage <menage@google.com> wrote:
>
>> This set of patches makes the Control Groups API more structured and
>> self-describing.
>>
>> 1) Allows control files to be associated with data types such as
>> "u64", "string", "map", etc. These types show up in a new cgroup.api
>> file in each cgroup directory, along with a user-readable
>> string. Files that use cgroup-provided data accessors have these file
>> types inferred automatically.
>>
>> 2) Moves various files in cpusets and the memory controller from using
>> custom-written file handlers to cgroup-defined handlers
>>
>> 3) Adds the "cgroup." prefix for existing cgroup-provided control
>> files (tasks, release_agent, releasable, notify_on_release). Given
>> than we've already had 2.6.24 go out without this prefix, I guess this
>> could be a little contentious - but it seems like a good move to
>> prevent name clashes in the future. (Note that this doesn't affect
>> mounting the legacy cpuset filesystem, since the compatibility layer
>> disables all prefixes when mounted with filesystem type "cpuset"). If
>> people object too strongly, we could just make this the case for *new*
>> cgroup API files, but I think this is a case where consistency would
>> be better than compatibility - I'd be surprised if anyone has written
>> major legacy apps yet that rely on 2.6.24 cgroup control file names.
>>
>
>
> Hi, I like this direction very much. thank you for your work.
> Self-describing cgroup.api file is a good idea!
>
> One request from me is add "mode" bit to cftype for allowing
> write-only/read-only files.
>
> Thanks,
> -Kame
>

I don't quite catch what you mean. Cgoup does support write-only/read-only
files. For a write-only file, just set .write and .write_uint to be NULL,
similar for a read-only file.

Do I miss something?

_____

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