

On Feb 19, 2008 1:57 PM, Paul Jackson <pj@sgi.com> wrote:

>
> Finally, it goes against the one thingie per file (at most, one scalar
> vector) that has worked well for us when tried.

Right, I like the idea of keeping things simple. But if you're going to accept that a vector is useful, then it seems reasonable that some other *simple* structured datatypes can be useful. An N-element key/value map (a la /proc/meminfo) is, I think, nicer than having to read values from N separate files.

>
> As to the motivations Paul M gives:
> 1) Avoid "an arbitrary mess of ad-hoc APIs":
> We can still do that, whether or not we "self-document" these
> API's in this manner.

We can, but this file makes it more clear what control files have a well-defined API and which are just returning some ad-hoc string.

I guess it's not essential, I just figured that if we had that information, it made sense to make it available to userspace. I guess I'm happy with dropping the actual exposed cgroup.api file for now as long as we can work towards reducing the number of control files that just return strings, and make use of the structured output such as read_uint() more.

> 2) binary APIs versus ASCII APIs:
> Well, I have an ASCII API bias, not surprising. But I'd
> suggest not doing things "in anticipation" of some future
> fuzzy binary API support. Wait until that day actually arrives.

I have a reasonably clear idea of how we can do the binary API.

That's mostly for a separate RFC. But for example, reading a map via the binary API would be able to just return a list values since the keys could be parsed once from the ascii map (provided that the subsystem guaranteed that the map keys and their order wouldn't change between reboots).

> 3) The memory controller currently has files with the "_in_bytes":
> The traditional way to handle this is Documentation and man
> pages; good enough for my granddad, good enough for me ;).

I've tried submitting patches to remove the in_bytes suffix and just rely on the documentation, and people didn't seem to like it ...

Paul

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