Subject: Re: How to draw values for /proc/stat
Posted by Glauber Costa on Mon, 12 Dec 2011 07:06:34 GMT
View Forum Message <> Reply to Message

On 12/12/2011 04:31 AM, KAMEZAWA Hiroyuki wrote:
> On Sun, 11 Dec 2011 15:50:56 +0100
> Glauber Costa<glommer@parallels.com> wrote:
>
>> On 12/09/2011 03:55 PM, Glauber Costa wrote:
>>> On 12/09/2011 12:03 PM, Peter Zijlstra wrote:
>>>> On Mon, 2011-12-05 at 07:32 -0200, Glauber Costa wrote:
>>>>> Hi,
>>>>>
>>>>> Specially Peter and Paul, but all the others:
>>>>>
>>>>> As you can see in https://lkml.org/lkml/2011/12/4/178, and in my answer
>>>>> to that, there is a question - one I've asked before but without that
>>>>> much of an audience - of whether /proc files read from process living on
>>>>> cgroups should display global or per-cgroup resources.
>>>>>
>>>>> In the past, I was arguing for a knob to control that, but I recently
>>>>> started to believe that a knob here will only overcomplicate matters:
>>>>> if you live in a cgroup, you should display only the resources you can
>>>>> possibly use. Global is for whoever is in the main cgroup.
>>>>>
>>>>> Now, it comes two questions:
>>>>> 1) Do you agree with that, for files like /proc/stat ? I think the most
>>>>> important part is to be consistent inside the system, regardless of what
>>>>> is done
>>>>
>>>> Personally I don't give a rats arse about (/proc vs) cgroups :-)
>>>> Currently /proc is unaffected by whatever cgroup you happen to be in and
>>>> that seems to make some sort of sense.
>>>>
>>>> Namespaces seem to be about limiting visibility, cgroups about
>>>> controlling resources.
>>>>
>>>> The two things are hopelessly disjoint atm, but I believe someone was
>>>> looking at this mess.
>>>
>>> I did take a look at this (if anyone else was, I'd like to know so we
>>> can share some ideas), but I am not convinced we should do anything to
>>> join them anymore. We virtualization people are to the best of my
>>> knowledge the only ones doing namespaces. Cgroups, OTOH, got a lot bigger.
>>>
>>> What I am mostly concerned about now, is how consistent they will be.
>>> /proc always being always global indeed does make sense, but my question
>>> still stands: if you live in a resource-controlled world, why should you

>>> even see resources you will never own ?
>>>
>>>
>>>> IOW a /proc namespace coupled to cgroup scope would do what you want.
>>>> Now my head hurts..
>>>
>>> Mine too. The idea is good, but too broad. Boils down to: How do you
>>> couple them? And none of the methods I thought about seemed to make any
>>> sense.
>>>
>>> If we really want to have the values in /proc being opted-in, I think
>>> Kamezawa's idea of a mount option is the winner so far.
>>>
>>
>> Ok:
>>
>> How about the following patch to achieve this ?
>
> Hmm, What I thought was mount option for procfs. Containers will mount its own
> /proc file systems. Do you have any pros. / cons. ?
> IIUC, cgroup can be mounted per subsystems. Then, options can be passed per
> subsystems. It's a mess but we don't need to bring this to procfs.
>
> How about
>
>   # mount -t procfs proc /container_root/proc -o cgroup_aware
>
> to show cgroup aware procfs ? I think this will be easy to be used with
> namespace/chroot, etc.
>

Don't think it works.

Because whoever mounts the proc filesystem, may not want to be isolated.
But we want him to be.

As an example from our usecase, procfs is mounted inside a container. We
can't assume the container is willing to cooperate. So we need to
establish this from the outside. We can of course force options to be
always added to a procfs mount if it comes from the container, but it is
way more messier than this.

per-cgroup knobs works fine for this because the container cannot
possibly see it or change it in any circumstance.
per-namespace would work as well, but then I don't see how to specify a
want/don't want flag in a sane way.