Subject: Re: Containers and /proc/sys/vm/drop_caches
Posted by Rob Landley on Sat, 08 Jan 2011 12:39:31 GMT
View Forum Message <> Reply to Message

On 01/07/2011 09:12 AM, Serge Hallyn wrote:
>> Changing ownership so a script can't open a file that it otherwise
>> could may cause scripts to fail when run in a container. Makes
>> the containers less transparent.
>
> While my goal next week is to make containers more transparent, the
> official stance from kernel summit a few years ago was: transparent
> containers are not a valid goal (as seen from kernel).

Do you have a reference for that? I'm still coming up to speed on all this. Trying to collect documentation...

>> A heavily loaded system that goes deep into swap without triggering
>> the OOM killer can become pretty useless. My home laptop with 2
>> gigs
>
> Isn't a cgroup that controls both memory and swap access the right
> answer to this?

There are other ways to work around it, sure. (It's yet to be proven that they do actually work better in resource constrained desktop environments under real-world load, but they seem very promising.)

I was just pointing out that this has seen some use as a recovery mechanism, slightly less drastic than the OOM killer. (Didn't say it was a _good_ use. Also, error avoidance and error recovery are different issues, and virtual memory is an inherently overcommitted resource domain.)

> (And do we have that now, btw?)

I think it's coming, rather than actually here. (I thought the beancounters stuff was OpenVZ, controlled by syscalls that the kernel developers rejected. Have resource constraints on anything other than scheduler made it into vanilla yet? If so, what's the UI to control them?)

By the way, from a UI perspective, most of the containers stuff I've seen so far is apparently aimed at big iron deployments (or attempts to make PC clusters look like mainframes, I.E. this "cloud" stuff). I'm glad to see more diverse uses of it, but one of the downsides of cobbling together a mechanism from a dozen different unrelated pieces of infrastructure (clone flags, cgroup filesystem, extra mount flags on proc and such so they behave differently) is that we need a lot of documentation/example code/libraries to make it easy to use. "You can do X" and "it's easy to reliably do X" have a gap that may take a while to close...

Rob

_____
Containers mailing list