

On Mon, Oct 30, 2006 at 05:08:03PM +0300, Pavel Emelianov wrote:

- > 1. One of the major configs ideas is that lifetime of
- > the objects is completely driven by userspace.
- > Resource controller shouldn't live as long as user
- > want. It "may", but not "must"! As you have seen from
- > our (beancounters) patches beancounters disappeared
- > as soon as the last reference was dropped. Removing
- > configs entries on beancounter's automatic destruction
- > is possible, but it breaks the logic of configs.

cpusets has a neat flag called `notify_on_release`. If set, some userspace agent is invoked when the last task exists from a cpuset.

Can't we use a similar flag as a configs file and (if set) invoke a userspace agent (to cleanup) upon last reference drop? How would this violate logic of configs?

- > 2. Having configs as the only interface doesn't allow
- > people having resource controll facility w/o configs.
- > Resource controller must not depend on any "feature".

One flexibility configs (and any fs-based interface) offers is, as Matt had pointed out sometime back, the ability to delage management of a sub-tree to a particular user (without requiring root permission).

For ex:

```

/
|
-----
| |
|  vatsa (70%) linux (20%)
|
-----
|   |   |
|   browser (10%)  compile (50%)  editor (10%)
```

In this, group 'vatsa' has been allotted 70% share of cpu. Also user 'vatsa' has been given permissions to manage this share as he wants. If the cpu controller supports hierarchy, user 'vatsa' can create further sub-groups (browser, compile ..etc) -without- requiring root access.

Also it is convenient to manipulate resource hierarchy/parameters thr a shell-script if it is fs-based.

> 3. Configfs may be easily implemented later as an additional  
> interface. I propose the following solution:

Ideally we should have one interface - either syscall or configfs - and not both.

Assuming your requirement of auto-deleting objects in configfs can be met thr' something similar to cpuset's `notify_on_release`, what other killer problem do you think configfs will pose?

> > - Should we have different groupings for different resources?  
>  
> This breaks the idea of groups isolation.

Sorry dont get you here. Are you saying we should support different grouping for different controllers?

> > - Support movement of all threads of a process from one group  
> > to another atomically?  
>  
> This is not a critical question. This is something that  
> has difference in

It can be a significant pain for some workloads. I have heard that workload management products often encounter processes with anywhere between 200-700 threads in a process. Moving all those threads one by one from user-space can suck.

--

Regards,  
vatsa

---