

Hi Balbir,

On 5/14/07, Balbir Singh <balbir@linux.vnet.ibm.com> wrote:

>
> This patch implements per container statistics infrastructure and re-uses
> code from the taskstats interface. A new set of container operations are
> registered with commands and attributes. It should be very easy to
> extend per container statistics, by adding members to the containerstats
> structure.

Sorry for the delay in looking at this. (I've been travelling a bit).

The basic idea of being able to get stats on a per-container basis seems good, but I've got some suggestions on the API/implementation:

- saving a mount pointer in the containerfsroot structure won't work because a hierarchy (superblock) can be mounted in more than one place, or even in zero places (if you unmount a hierarchy with active containers, the superblock and the containers stay active). Also it might be possible to move a mounted container hierarchy via mount --move, although I've not actually tried that.

- a cleaner way to pass in a container id would be to pass a file descriptor on a container directory. The dentry associated with this fd would unambiguously identify the hierarchy and the container, so then even if we didn't maintain a per-container task list, the for_each_thread() loop would involve just a single comparison per task to see if the task was in the desired container.

- if we're trying to integrate this with taskstats, then it would be nice to support retrieving all the other taskstats values (where they make sense) on a per-container basis (in the same way that we can currently request them on a per-thread or a per-process basis). i.e. don't create a new container_taskstats structure, but instead augment the current taskstats structure, and allow the user to retrieve the aggregate of that for the entire container. Similarly, being able to get taskstats notifications based on the container memberships of a task might be nice.

Paul
