

On Fri, Apr 18, 2008 at 10:35 PM, Balbir Singh
<balbir@linux.vnet.ibm.com> wrote:

- >
- > 1. We need to hold cgroup_mutex while walking through the children
- > in reclaim. We need to figure out the best way to do so. Should
- > cgroups provide a helper function/macro for it?

There's already a function, cgroup_lock(). But it would be nice to avoid such a heavy locking here, particularly since memory allocations can occur with cgroup_mutex held, which could lead to a nasty deadlock if the allocation triggered reclaim.

One of the things that I've been considering was to put the parent/child/sibling hierarchy explicitly in cgroup_subsys_state. This would give subsystems their own copy to refer to, and could use their own internal locking to synchronize with callbacks from cgroups that might change the hierarchy. Cpusets could make use of this too, since it has to traverse hierarchies sometimes.

- > 2. Do not allow children to have a limit greater than their parents.
- > 3. Allow the user to select if hierarchial support is required

My thoughts on this would be:

1) Never attach a first-level child's counter to its parent. As Yamamoto points out, otherwise we end up with extra global operations whenever any cgroup allocates or frees memory. Limiting the total system memory used by all user processes doesn't seem to be something that people are going to generally want to do, and if they really do want to they can just create a non-root child and move the whole system into that.

The one big advantage that you currently get from having all first-level children be attached to the root is that the reclaim logic automatically scans other groups when it reaches the top-level - but I think that can be provided as a special-case in the reclaim traversal, avoiding the overhead of hitting the root cgroup that we have in this patch.

2) Always attach other children's counters to their parents - if the user didn't want a hierarchy, they could create a flat grouping rather than nested groupings.

Paul

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