Contribution ID: **409** Type: **not specified**

Solving Zombie Cgroups - Simple Fix or Smart Solution?

Friday 12 December 2025 10:45 (45 minutes)

For years, the Linux kernel has been troubled by "zombie memory cgroups." When a memory cgroup is destroyed, shared file pages often remain charged to it, preventing the kernel from freeing the cgroup's metadata. Over time, this creates a slow but steady memory leak in environments where cgroups are frequently created and destroyed.

The community is working on a fix by reparenting these stuck pages to the parent memory cgroup, allowing the metadata of the dying cgroup to be released. At this stage of upstreaming, the priority is simplicity and correctness, as attempts to chase a more sophisticated solution have been frustrated by increased complexity. The simple solution works, but it has a disadvantage: reparented pages may lose their hotness information. They may be placed into young or old LRU lists regardless of their actual hotness, potentially leading to suboptimal reclamation.

In this Birds of a Feather session, we would like to hear from you: are there realistic workloads where memory cgroups with very large memory footprints are created and destroyed frequently enough that this loss of accuracy would cause real problems for reclamation performance, warranting additional complexity?

Our goal is not to present a finished solution, but to find out whether the added complexity of a more accurate algorithm is truly justified. We need your input: your workloads, insights, performance data, and real-world concerns to help determine the right path forward.

Primary author: YOO, Harry (Oracle)

Co-author: BABULAL, Kamalesh (Oracle)

Presenters: YOO, Harry (Oracle); BABULAL, Kamalesh (Oracle)

Session Classification: Birds of a Feather (BoF)

Track Classification: Birds of a Feather (BoF)