

Hello all,

So after thinking a lot about the last round of kmem memcg patches, this is what I managed to come up with. I am not sending the whole series for two reasons:

- 1) It still have a nasty destruction bug, and a slab heisenbug (slub seems to be working as flawlessly as before), and I'd like to gather your comments early on this approach
- 2) The rest of the series doesn't change *that* much. Most patches are to some extent touched, but it's mainly to adapt to those four, which I consider to be the core changes between last series. So you can focus on these, and not be distracted by the surrounding churn.

The main difference here is that as suggested by Cristoph, I am hooking at the page allocator. It is, indeed looser than before. But I still keep objects from the same cgroup in the same page most of the time. I guarantee that by using the same dispatch mechanism as before to select a particular per-memcg cache, but I now assume the process doing the dispatch will be the same doing the page allocation.

The only situation this does not hold true, is when a task moves cgroup *between those two events*. So first of all, this is fixable. One can have a reaper, a check while moving, a match check after the page is allocated. But also, this is the kind of loose accounting I don't care too much about, since this is expected to be a rare event, one I particularly don't care about, and more importantly, it won't break anything.

Let me know what you people think of this approach. In terms of meddling with the internals of the caches, it is way less invasive than before.

Glauber Costa (4):

memcg: kmem controller dispatch infrastructure
Add a __GFP_SLABMEMCG flag
don't do __ClearPageSlab before freeing slab page.
mm: Allocate kernel pages to the right memcg

```
include/linux/gfp.h      | 4 +-  
include/linux/memcontrol.h | 72 ++++++++  
include/linux/page-flags.h | 2 +-  
include/linux/slub_def.h | 15 +-  
init/Kconfig            | 2 +-  
mm/memcontrol.c         | 358 ++++++
```

```
mm/page_alloc.c      | 16 +-  
mm/slab.c            | 9 +-  
mm/slob.c            | 1 -  
mm/slub.c            | 2 +-  
10 files changed, 464 insertions(+), 17 deletions(-)
```

--
1.7.10.2
