Subject: [PATCH v3 00/28] kmem limitation for memcg
Posted by Glauber Costa on Fri, 25 May 2012 13:03:20 GMT
View Forum Message <> Reply to Message

Hello All,

This is my new take for the memcg kmem accounting. This should merge
all of the previous comments from you, plus fix a bunch of bugs.

At this point, I consider the series pretty mature. Since last submission
2 weeks ago, I focused on broadening the testing coverage. Some bugs were
fixed, but that of course doesn't mean no bugs exist.

I believe some of the early patches here are already in some trees around.
I don't know who should pick this, so if everyone agrees with what's in here,
please just ack them and tell me which tree I should aim for (-mm? Hocko's?)
and I'll rebase it.

I should point out again that most, if not all, of the code in the caches
are wrapped in static_key areas, meaning they will be completely patched out
until the first limit is set. Enabling and disabling of static_keys incorporate
the last fixes for sock memcg, and should be pretty robust.

I also put a lot of effort, as you will all see, in the proper separation
of the patches, so the review process is made as easy as the complexity of
the work allows to.

[v3]
 * fixed lockdep bugs in slab (ordering of get_online_cpus() vs slab_mutex)
 * improved style in slab and slub with less #ifdefs in-code
 * tested and fixed hierarchical accounting (memcg: propagate kmem limiting...)
 * some more small bug fixes
 * No longer using res_counter_charge_nofail for GFP_NOFAIL submissions. Those
   go to the root memcg directly.
 * reordered tests in mem_cgroup_get_kmem_cache so we exit even earlier for
   tasks in root memcg
 * no more memcg state for slub initialization
 * do_tune_cpucache will always (only after FULL) propagate to children when
   they exist.
 * slab itself will destroy the kmem_cache string for chained caches, so we
   don't need to bother with consistency between them.
 * other minor issues
[v2]
 * memcgs can be properly removed.
 * We are not charging based on current->mm->owner instead of current
 * kmem_large allocations for slub got some fixes, specially for the free case
 * A cache that is registered can be properly removed (common module case)
   even if it spans memcg children. Slab had some code for that, now it works

well with both
* A new mechanism for skipping allocations is proposed (patch posted
separately already). Now instead of having kmalloc_no_account, we mark
a region as non-accountable for memcg.

Glauber Costa (25):
slab: move FULL state transition to an initcall
memcg: Always free struct memcg through schedule_work()
slab: rename gfpflags to allocflags
slab: use obj_size field of struct kmem_cache when not debugging
memcg: change defines to an enum
res_counter: don't force return value checking in
res_counter_charge_nofail
kmem slab accounting basic infrastructure
slab/slub: struct memcg_params
slub: consider a memcg parameter in kmem_create_cache
slab: pass memcg parameter to kmem_cache_create
slub: create duplicate cache
slab: create duplicate cache
slub: always get the cache from its page in kfree
memcg: kmem controller charge/uncharge infrastructure
skip memcg kmem allocations in specified code regions
slub: charge allocation to a memcg
slab: per-memcg accounting of slab caches
memcg: disable kmem code when not in use.
memcg: destroy memcg caches
memcg/slub: shrink dead caches
slab: Track all the memcg children of a kmem_cache.
slub: create slabinfo file for memcg
slub: track all children of a kmem cache
memcg: propagate kmem limiting information to children
Documentation: add documentation for slab tracker for memcg

Suleiman Souhlal (3):
memcg: Make it possible to use the stock for more than one page.
memcg: Reclaim when more than one page needed.
memcg: Per-memcg memory.kmem.slabinfo file.

Documentation/cgroups/memory.txt | 33 ++
include/linux/memcontrol.h | 101 +++++
include/linux/res_counter.h | 2 +-
include/linux/sched.h | 1 +
include/linux/slab.h | 32 ++
include/linux/slab_def.h | 79 ++++-
include/linux/slub_def.h | 68 ++-
init/Kconfig | 2 +-
mm/memcontrol.c | 897 ++--
mm/slab.c | 423 +++++++++++++---

mm/slub.c | 282 ++++++++++-
11 files changed, 1787 insertions(+), 133 deletions(-)

--
1.7.7.6