Subject: [PATCH v2 00/29] kmem limitation for memcg
Posted by Glauber Costa on Fri, 11 May 2012 17:44:02 GMT
View Forum Message <> Reply to Message

Hello All,

This is my new take for the memcg kmem accounting.
At this point, I consider the series pretty mature - although of course,
bugs are always there...

As a disclaimer, however, I must say that the slub code is much more stressed
by me, since I know it better. If you have no more objections to the concepts
presented, the remaining edges can probably be polished in a rc cycle,
at the maintainers discretion, of course.

Otherwise, I'll be happy to address any concerns of yours.

Since last submission:

 * 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.

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.

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.

Frederic Weisbecker (1):
  cgroups: ability to stop res charge propagation on bounded ancestor

Glauber Costa (24):
  slab: dup name string
  slub: fix slab_state for slub
  memcg: Always free struct memcg through schedule_work()
  slub: always get the cache from its page in kfree
  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
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
slub: create slabinfo file for memcg
slub: track all children of a kmem cache
Documentation: add documentation for slab tracker for memcg

Suleiman Souhlal (4):
memcg: Make it possible to use the stock for more than one page.
memcg: Reclaim when more than one page needed.
memcg: Track all the memcg children of a kmem_cache.
memcg: Per-memcg memory.kmem.slabinfo file.

Documentation/cgroups/memory.txt | 33 ++
Documentation/cgroups/resource_counter.txt | 18 +-
include/linux/memcontrol.h | 88 ++++
include/linux/res_counter.h | 23 +-
include/linux/sched.h | 1 +
include/linux/slab.h | 29 +
include/linux/slab_def.h | 72 +++-
include/linux/slub_def.h | 51 ++-
init/Kconfig | 2 +-
kernel/res_counter.c | 13 +-
mm/memcontrol.c | 773 ++++++++++++++++++++++++++++++--
mm/slab.c | 394 ++++++++++++---
mm/slub.c | 298 ++++++++++-
13 files changed, 1658 insertions(+), 137 deletions(-)

--
1.7.7.6