Subject: [PATCH v5 00/18] slab accounting for memcg
Posted by Glauber Costa on Fri, 19 Oct 2012 14:20:24 GMT
View Forum Message <> Reply to Message

Note: This is basically the same as v4. During this week, I made some changes
to this series in advance based on the feedback I had in the kmemcg-stack
last submission. Although the last series was not yet extensively reviewed,
I opted for sending this out so you guys have the most up2date code to review.
So please review this one instead.

This is a followup to the previous kmem series. I divided them logically
so it gets easier for reviewers. But I believe they are ready to be merged
together (although we can do a two-pass merge if people would prefer)

Throwaway git tree found at:

git://git.kernel.org/pub/scm/linux/kernel/git/glommer/memcg. git kmemcg-slab

v5:
* code reorganization, name changes, etc.
v4:
* no more messing with the cache name after destruction: aggregated figures
are shown in /proc/slabinfo.
* memory.kmem.slabinfo file with memcg-specific cache information during its
lifespan.
* full slub attribute propagation.
* reusing the standard workqueue mechanism.
* cache-side indexing, instead of memcg-side indexing. The memcg css_id serves
as an index, and we don't need extra indexes for that.
* struct memcg_cache_params no longer bundled in struct kmem_cache: We now will
have only a pointer in the struct, allowing memory consumption when disable to
fall down ever further.

Patches need to be adjusted to cope with those changes, but other than that,
look the same - just a lot simpler.

I also put quite some effort to overcome my writing disability and get some
decent changelogs in place.

For a detailed explanation about this whole effort, please refer to my previous
post (https://lkml.org/lkml/2012/10/8/119)

Glauber Costa (18):
move slabinfo processing to slab_common.c
move print_slabinfo_header to slab_common.c
sl[au]b: process slabinfo_show in common code
slab: don't preemptively remove element from list in cache destroy
slab/slub: struct memcg_params

consider a memcg parameter in kmem_create_cache
Allocate memory for memcg caches whenever a new memcg appears
memcg: infrastructure to match an allocation to the right cache
memcg: skip memcg kmem allocations in specified code regions
sl[au]b: always get the cache from its page in kfree
sl[au]b: Allocate objects from memcg cache
memcg: destroy memcg caches
memcg/sl[au]b Track all the memcg children of a kmem_cache.
memcg/sl[au]b: shrink dead caches
Aggregate memcg cache values in slabinfo
slab: propagate tunables values
slub: slub-specific propagation changes.
Add slab-specific documentation about the kmem controller

```
Documentation/cgroups/memory.txt |   7 +
include/linux/memcontrol.h       |  86 ++++++
include/linux/sched.h            |   1 +
include/linux/slab.h             |  46 +++
include/linux/slab_def.h         |   3 +
include/linux/slub_def.h         |  19 +-
init/Kconfig                     |   2 +-
mm/memcontrol.c                  | 613 ++++++++++++++++++++++++++++++++++++++++--
mm/slab.c                        | 188 ++++++------
mm/slab.h                        | 132 ++++++++-
mm/slab_common.c                 | 217 +++++++++++++-
mm/slub.c                        | 195 ++++++++----
12 files changed, 1294 insertions(+), 215 deletions(-)
```

--
1.7.11.7