Subject: [PATCH v4 22/25] slab: slab-specific propagation changes.
Posted by Glauber Costa on Mon, 18 Jun 2012 10:28:15 GMT
View Forum Message <> Reply to Message

When a parent cache does tune_cpucache, we need to propagate that to the
children as well. For that, we unfortunately need to tap into the slab core.

Signed-off-by: Glauber Costa <glommer@parallels.com>
CC: Christoph Lameter <cl@linux.com>
CC: Pekka Enberg <penberg@cs.helsinki.fi>
CC: Michal Hocko <mhocko@suse.cz>
CC: Kamezawa Hiroyuki <kamezawa.hiroyu@jp.fujitsu.com>
CC: Johannes Weiner <hannes@cmpxchg.org>
CC: Suleiman Souhlal <suleiman@google.com>
---
 mm/slab.c | 28 ++++++++++++++++++++++++++-
 mm/slab_common.c | 1 +
 2 files changed, 28 insertions(+), 1 deletion(-)

diff --git a/mm/slab.c b/mm/slab.c
index 4951c81..c280dc6 100644
--- a/mm/slab.c
+++ b/mm/slab.c
@@ -3874,7 +3874,7 @@ static void do_ccupdate_local(void *info)
 }

 /* Always called with the slab_mutex held */
-static int do_tune_cpucache(struct kmem_cache *cachep, int limit,
+static int __do_tune_cpucache(struct kmem_cache *cachep, int limit,
     int batchcount, int shared, gfp_t gfp)
 {
  struct ccupdate_struct *new;
@@ -3917,6 +3917,32 @@ static int do_tune_cpucache(struct kmem_cache *cachep, int limit,
  return alloc_kmemlist(cachep, gfp);
 }

+static int do_tune_cpucache(struct kmem_cache *cachep, int limit,
+   int batchcount, int shared, gfp_t gfp)
+{
+ int ret;
+#ifdef CONFIG_CGROUP_MEM_RES_CTLR_KMEM
+ struct kmem_cache *c;
+ struct mem_cgroup_cache_params *p;
+#endif
+
+ ret = __do_tune_cpucache(cachep, limit, batchcount, shared, gfp);
+#ifdef CONFIG_CGROUP_MEM_RES_CTLR_KMEM
+ if (slab_state < FULL)

```
+  return ret;
+
+ if ((ret < 0) || (cachep->memcg_params.id == -1))
+  return ret;
+
+ list_for_each_entry(p, &cachep->memcg_params.sibling_list, sibling_list) {
+  c = container_of(p, struct kmem_cache, memcg_params);
+  /* return value determined by the parent cache only */
+  __do_tune_cpucache(c, limit, batchcount, shared, gfp);
+ }
+#endif
+ return ret;
+}
+
 /* Called with slab_mutex held always */
 static int enable_cpucache(struct kmem_cache *cachep, gfp_t gfp)
 {
diff --git a/mm/slab_common.c b/mm/slab_common.c
index b424b28..a8557e8 100644
--- a/mm/slab_common.c
+++ b/mm/slab_common.c
@@ -155,6 +155,7 @@ kmem_cache_create_memcg(struct mem_cgroup *memcg, const char
*name, size_t size,
  s->flags = flags;
  s->align = calculate_alignment(flags, align, size);
 #ifdef CONFIG_CGROUP_MEM_RES_CTLR_KMEM
+ s->memcg_params.id = -1; /* not registered yet */
  s->memcg_params.memcg = memcg;
  s->memcg_params.parent = parent_cache;
 #endif
--
1.7.10.2
```