Subject: [PATCH v3 14/16] slub: slub-specific propagation changes.
Posted by Glauber Costa on Tue, 18 Sep 2012 14:12:08 GMT
View Forum Message <> Reply to Message

When a parent cache changes a sysfs attr, we need to propagate that to
the children as well. For that, we unfortunately need to tap into the
slub 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/slub.c | 16 ++++++++++++++++
 1 file changed, 16 insertions(+)

```
diff --git a/mm/slub.c b/mm/slub.c
index f90f612..0b68d15 100644
--- a/mm/slub.c
+++ b/mm/slub.c
@@ -5174,6 +5174,10 @@ static ssize_t slab_attr_store(struct kobject *kobj,
  struct slab_attribute *attribute;
  struct kmem_cache *s;
  int err;
+#ifdef CONFIG_MEMCG_KMEM
+ struct kmem_cache *c;
+ struct mem_cgroup_cache_params *p;
+#endif

  attribute = to_slab_attr(attr);
  s = to_slab(kobj);
@@ -5182,7 +5186,19 @@ static ssize_t slab_attr_store(struct kobject *kobj,
   return -EIO;

  err = attribute->store(s, buf, len);
+#ifdef CONFIG_MEMCG_KMEM
+ if (slab_state < FULL)
+  return err;

+ if ((err < 0) || (s->memcg_params.id == -1))
+  return err;
+
+ list_for_each_entry(p, &s->memcg_params.sibling_list, sibling_list) {
+  c = container_of(p, struct kmem_cache, memcg_params);
+  /* return value determined by the parent cache only */
```

```
+  attribute->store(c, buf, len);
+ }
+#endif
  return err;
 }

--
1.7.11.4
```