Subject: [RFC][PATCH 3/3] Tune caches to be accountable or not
Posted by Pavel Emelianov on Thu, 13 Sep 2007 09:16:05 GMT
View Forum Message <> Reply to Message

The /sys/slab/<name>/cache_account attribute controls
whether the cache <name> is to be accounted or not.

For the reasons described in the zeroth letter kmalloc
caches are excluded and are not allowed to be merged.

By default no caches are accountable. Simply make
  # echo -n 1 > /sys/slab/<name>cache_account
to turn accounting on.

Other caches can be accountable, but if we turn accounting
on on some cache and this cache is merged with some other,
this "other" will be accountable as well. We can solve this
by disabling of cache merging, but I'd prefer to know
Christoph's opinion first.

Turning the accounting off is possible only when this cache
is empty.

Signed-off-by: Pavel Emelyanov <xemul@openvz.org>

---

 mm/slub.c | 51 ++
 1 files changed, 51 insertions(+)

diff --git a/mm/slub.c b/mm/slub.c
index 113df81..baf8da4 100644
--- a/mm/slub.c
+++ b/mm/slub.c
@@ -2878,6 +2878,16 @@ static int slab_unmergeable(struct kmem_
  if (s->refcount < 0)
   return 1;

+#ifdef CONFIG_CONTAINER_KMEM
+ /*
+ * many caches that can be accountable are usually merged with
+ * kmalloc caches, which are disabled for accounting for a while
+ */
+
+ if (is_kmalloc_cache(s))
+ return 1;
+#endif
+

```
  return 0;
 }

@@ -3842,6 +3852,44 @@ static ssize_t defrag_ratio_store(struct
 SLAB_ATTR(defrag_ratio);
 #endif

+#ifdef CONFIG_CONTAINER_KMEM
+static ssize_t cache_account_show(struct kmem_cache *s, char *buf)
+{
+ return sprintf(buf, "%d\n", !!(s->flags & SLAB_CHARGE));
+}
+
+static ssize_t cache_account_store(struct kmem_cache *s,
+   const char *buf, size_t length)
+{
+ if (buf[0] == '1') {
+   if (is_kmalloc_cache(s))
+     /*
+      * cannot just make these caches accountable
+      */
+     return -EINVAL;
+
+   s->flags |= SLAB_CHARGE;
+   return length;
+ }
+
+ if (buf[0] == '0') {
+   if (any_slab_objects(s))
+     /*
+      * we cannot turn this off because of the
+      * full slabs cannot be found in this case
+      */
+     return -EBUSY;
+
+   s->flags &= ~SLAB_CHARGE;
+   return length;
+ }
+
+ return -EINVAL;
+}
+
+SLAB_ATTR(cache_account);
+#endif
+
 static struct attribute * slab_attrs[] = {
   &slab_size_attr.attr,
   &object_size_attr.attr,
```

```
@@ -3872,6 +3920,9 @@ static struct attribute * slab_attrs[] =
 #ifdef CONFIG_NUMA
  &defrag_ratio_attr.attr,
 #endif
+#ifdef CONFIG_CONTAINER_KMEM
+ &cache_account_attr.attr,
+#endif
  NULL
 };
```