Subject: [PATCH 3/5] Switch caches notification dynamically
Posted by Pavel Emelianov on Fri, 21 Sep 2007 09:22:03 GMT
View Forum Message <> Reply to Message

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

For the reasons described before kmalloc caches cannot
be turned on.

By default no caches are accountable. Simply make
  # echo -n 1 > /sys/slab/<name>cache_notify
to turn notification of this cache on.

If we turn accounting on on some cache and this cache
is merged with some other, this "other" will be notified
as well. We can solve this by disabling of cache merging.

Turning the notification off is possible only when this
cache is empty. The reason for this is that the pages,
that are full of objects are not linked in any list, so
we wouldn't be able to walk these pages and notify others
that these objects are no longer tracked.

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

```
diff --git a/mm/slub.c b/mm/slub.c
index ac4f157..b5af598 100644
--- a/mm/slub.c
+++ b/mm/slub.c
@@ -3775,6 +3917,36 @@ static ssize_t defrag_ratio_store(struct
 SLAB_ATTR(defrag_ratio);
 #endif

+static ssize_t cache_notify_show(struct kmem_cache *s, char *buf)
+{
+ return sprintf(buf, "%d\n", !!(s->flags & SLAB_NOTIFY));
+}
+
+static ssize_t cache_notify_store(struct kmem_cache *s,
+  const char *buf, size_t length)
+{
+ if (buf[0] == '1') {
+  s->flags |= SLAB_NOTIFY;
+  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_NOTIFY;
+  return length;
+ }
+
+ return -EINVAL;
+}
+
+SLAB_ATTR(cache_notify);
+
 static struct attribute * slab_attrs[] = {
  &slab_size_attr.attr,
  &object_size_attr.attr,
@@ -3805,6 +3984,7 @@ static struct attribute * slab_attrs[] =
 #ifdef CONFIG_NUMA
  &defrag_ratio_attr.attr,
 #endif
+ &cache_notify_attr.attr,
  NULL
 };
```