

---

Subject: [PATCH 2/4] Switch caches notification dynamically  
Posted by [Pavel Emelianov](#) on Mon, 17 Sep 2007 12:30:45 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, but maybe we can do it some other way.

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 Emelianov <xemul@openvz.org>

---

mm/slub.c | 45 +++  
1 files changed, 45 insertions(+)

diff --git a/mm/slub.c b/mm/slub.c  
index 1802645..bfb7c21 100644

--- a/mm/slub.c

+++ b/mm/slub.c

@@ -2338,6 +2440,14 @@ EXPORT\_SYMBOL(kmem\_cache\_destroy);  
struct kmem\_cache kmalloc\_caches[PAGE\_SHIFT] \_\_cacheline\_aligned;  
EXPORT\_SYMBOL(kmalloc\_caches);

+static inline int is\_kmalloc\_cache(struct kmem\_cache \*s)  
+{  
+ int km\_idx;  
+  
+ km\_idx = s - kmalloc\_caches;  
+ return km\_idx >= 0 && km\_idx < ARRAY\_SIZE(kmalloc\_caches);  
+}  
+

```

#ifdef CONFIG_ZONE_DMA
static struct kmem_cache *kmalloc_caches_dma[PAGE_SHIFT];
#endif
@@ -3753,6 +3874,42 @@ 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') {
+ if (is_kmalloc_cache(s))
+ /*
+  * cannot just make these caches accountable
+  */
+ return -EINVAL;
+
+ 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,
@@ -3783,6 +3940,7 @@ static struct attribute * slab_attrs[] =
#ifdef CONFIG_NUMA

```

```
&defrag_ratio_attr.attr,  
#endif  
+ &cache_notify_attr.attr,  
  NULL  
};
```

---