
Subject: [PATCH v5 05/18] slab/slub: struct memcg_params
Posted by [Glauber Costa](#) on Fri, 19 Oct 2012 14:20:29 GMT
[View Forum Message](#) <> [Reply to Message](#)

For the kmem slab controller, we need to record some extra information in the kmem_cache structure.

Signed-off-by: Glauber Costa <glommer@parallels.com>
Signed-off-by: Suleiman Souhlal <suleiman@google.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: Tejun Heo <tj@kernel.org>

include/linux/slab.h | 25 +++++
include/linux/slab_def.h | 3 +++
include/linux/slub_def.h | 3 +++
mm/slab.h | 13 +++++
4 files changed, 44 insertions(+)

```
diff --git a/include/linux/slab.h b/include/linux/slab.h
index 0dd2dfa..e4ea48a 100644
--- a/include/linux/slab.h
+++ b/include/linux/slab.h
@@ -177,6 +177,31 @@ unsigned int kmem_cache_size(struct kmem_cache *);
#define ARCH_SLAB_MINALIGN __alignof__(unsigned long long)
#endif
```

```
+#include <linux/workqueue.h>
+/*
+ * This is the main placeholder for memcg-related information in kmem caches.
+ * struct kmem_cache will hold a pointer to it, so the memory cost while
+ * disabled is 1 pointer. The runtime cost while enabled, gets bigger than it
+ * would otherwise be if that would be bundled in kmem_cache: we'll need an
+ * extra pointer chase. But the trade off clearly lays in favor of not
+ * penalizing non-users.
+ *
+ * Both the root cache and the child caches will have it. For the root cache,
+ * this will hold a dynamically allocated array large enough to hold
+ * information about the currently limited memcgs in the system.
+ *
+ * Child caches will hold extra metadata needed for its operation. Fields are:
+ *
+ * @memcg: pointer to the memcg this cache belongs to
+ */
+struct memcg_cache_params {
```



```
+static inline bool is_root_cache(struct kmem_cache *s)
+{
+ return !s->memcg_params || s->memcg_params->is_root_cache;
+}
+
+##else
+static inline bool is_root_cache(struct kmem_cache *s)
+{
+ return true;
+}
+
+##endif
+endif
+
+
+1.7.11.7
```