
Subject: [PATCH v3 25/28] slub: create slabinfo file for memcg
Posted by [Glauber Costa](#) on Fri, 25 May 2012 13:03:45 GMT
[View Forum Message](#) <> [Reply to Message](#)

This patch implements mem_cgroup_slabinfo() for the slub.
With that, we can also probe the used caches for it.

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 | 27 ++++++++++++++++++++++++++++++++++++++
1 files changed, 27 insertions(+), 0 deletions(-)

diff --git a/mm/slub.c b/mm/slub.c
index c8a8cab..4c29e5f 100644
--- a/mm/slub.c
+++ b/mm/slub.c
@@ -4113,6 +4113,33 @@ struct kmem_cache *kmem_cache_dup(struct mem_cgroup *memcg,

int mem_cgroup_slabinfo(struct mem_cgroup *memcg, struct seq_file *m)
{
+ struct kmem_cache *s;
+ int node;
+ unsigned long nr_objs = 0;
+ unsigned long nr_free = 0;
+
+ seq_printf(m, "# name          <active_objs> <num_objs> <objsize>\n");
+
+ down_read(&slub_lock);
+ list_for_each_entry(s, &slab_caches, list) {
+ if (s->memcg_params.memcg != memcg)
+ continue;
+
+ for_each_online_node(node) {
+ struct kmem_cache_node *n = get_node(s, node);
+
+ if (!n)
+ continue;
+
+ nr_objs += atomic_long_read(&n->total_objects);
+ nr_free += count_partial(n, count_free);
+ }
+
+ }
```

```
+ seq_printf(m, "%-17s %6lu %6lu %6u\n", s->name,  
+   nr_objs - nr_free, nr_objs, s->size);  
+ }  
+ up_read(&slub_lock);  
+  
+   return 0;  
+ }  
+ #endif  
+ --  
+ 1.7.7.6
```
