
Subject: [PATCH v5 08/10] Display current tcp memory allocation in kmem cgroup
Posted by [Glauber Costa](#) on Mon, 07 Nov 2011 15:26:33 GMT

[View Forum Message](#) <> [Reply to Message](#)

This patch introduces kmem.tcp.failcnt file, living in the kmem_cgroup filesystem. Following the pattern in the other memcg resources, this files keeps a counter of how many times allocation failed due to limits being hit in this cgroup. The root cgroup will always show a failcnt of 0.

Signed-off-by: Glauber Costa <glommer@parallels.com>
CC: David S. Miller <davem@davemloft.net>
CC: Hiroyouki Kamezawa <kamezawa.hiroyu@jp.fujitsu.com>
CC: Eric W. Biederman <ebiederm@xmission.com>

```
mm/memcontrol.c | 13 ++++++++
1 files changed, 13 insertions(+), 0 deletions(-)
```

```
diff --git a/mm/memcontrol.c b/mm/memcontrol.c
index 51b5a55..9394224 100644
```

```
--- a/mm/memcontrol.c
```

```
+++ b/mm/memcontrol.c
```

```
@@ -517,6 +517,7 @@ static int mem_cgroup_write(struct cgroup *cont, struct cftype *cft,
    const char *buffer);
```

```
static u64 mem_cgroup_read(struct cgroup *cont, struct cftype *cft);
+static int mem_cgroup_reset(struct cgroup *cont, unsigned int event);
```

```
/*
 * We need those things internally in pages, so don't reuse
 * mem_cgroup_{read,write}
```

```
@@ -533,6 +534,12 @@ static struct cftype tcp_files[] = {
    .read_u64 = mem_cgroup_read,
    .private = MEMFILE_PRIVATE(_KMEM_TCP, RES_USAGE),
    },
```

```
+ {
+ .name = "kmem.tcp.failcnt",
+ .private = MEMFILE_PRIVATE(_KMEM_TCP, RES_FAILCNT),
+ .trigger = mem_cgroup_reset,
+ .read_u64 = mem_cgroup_read,
+ },
};
```

```
static void tcp_create_cgroup(struct mem_cgroup *cg, struct cgroup_subsys *ss)
```

```
@@ -4134,6 +4141,8 @@ static u64 mem_cgroup_read(struct cgroup *cont, struct cftype *cft)
    if (mem_cgroup_is_root(mem)) {
        if (name == RES_USAGE)
            val = atomic_long_read(&tcp_memory_allocated) << PAGE_SHIFT;
+    else if (name == RES_FAILCNT)
```

```
+ val = 0;
  else
    val = RESOURCE_MAX;
  } else
@@ -4251,6 +4260,10 @@ static int mem_cgroup_reset(struct cgroup *cont, unsigned int event)
  case RES_FAILCNT:
    if (type == _MEM)
      res_counter_reset_failcnt(&mem->res);
+#if defined(CONFIG_CGROUP_MEM_RES_CTLR_KMEM) && defined(CONFIG_INET)
+ else if (type == _KMEM_TCP)
+ res_counter_reset_failcnt(&mem->tcp.tcp_memory_allocated);
+#endif
  else
    res_counter_reset_failcnt(&mem->memsw);
  break;
--
1.7.6.4
```
