
Subject: [PATCH 6/7] track dcache per-memcg
Posted by [Glauber Costa](#) on Tue, 21 Feb 2012 11:34:38 GMT
[View Forum Message](#) <> [Reply to Message](#)

This patch allows to track kernel memory used by dentry caches in the memory controller. It uses the infrastructure already laid down, and register the dcache as the first users of it.

A new cache is created for that purpose, and new allocations coming from tasks belonging to that cgroup will be serviced from the new cache.

Signed-off-by: Glauber Costa <glommer@parallels.com>
CC: Kirill A. Shutemov <kirill@shutemov.name>
CC: Greg Thelen <gthelen@google.com>
CC: Johannes Weiner <jweiner@redhat.com>
CC: Michal Hocko <mhocko@suse.cz>
CC: Hiroyouki Kamezawa <kamezawa.hiroyu@jp.fujitsu.com>
CC: Paul Turner <pjt@google.com>
CC: Frederic Weisbecker <fweisbec@gmail.com>
CC: Dave Chinner <david@fromorbit.com>

```
fs/dcache.c      | 38 ++++++-----
include/linux/dcache.h | 3 +++
2 files changed, 36 insertions(+), 5 deletions(-)
```

```
diff --git a/fs/dcache.c b/fs/dcache.c
index 16a53cc..a452c19 100644
```

```
--- a/fs/dcache.c
```

```
+++ b/fs/dcache.c
```

```
@@ -86,7 +86,7 @@ __cacheline_aligned_in_smp DEFINE_SEQLOCK(rename_lock);
```

```
EXPORT_SYMBOL(rename_lock);
```

```
-static struct kmem_cache *dentry_cache __read_mostly;
```

```
+static struct memcg_kmem_cache dentry_cache __read_mostly;
```

```
/*
```

```
 * This is the single most critical data structure when it comes
```

```
@@ -144,7 +144,7 @@ static void __d_free(struct rcu_head *head)
```

```
 WARN_ON(!list_empty(&dentry->d_alias));
```

```
 if (dname_external(dentry))
```

```
   kfree(dentry->d_name.name);
```

```
- kmem_cache_free(dentry_cache, dentry);
```

```
+ memcg_kmem_cache_free(dentry->d_cache->cache, dentry);
```

```
 }
```

```
/*
```

```

@@ -234,6 +234,7 @@ static void dentry_lru_add(struct dentry *dentry)
    if (list_empty(&dentry->d_lru)) {
        spin_lock(&dcache_lru_lock);
        list_add(&dentry->d_lru, &dentry->d_sb->s_dentry_lru);
+   dentry->d_cache->nr_objects++;
        dentry->d_sb->s_nr_dentry_unused++;
        dentry_stat.nr_unused++;
        spin_unlock(&dcache_lru_lock);
@@ -1178,6 +1179,21 @@ void shrink_dcache_parent(struct dentry * parent)
    }
    EXPORT_SYMBOL(shrink_dcache_parent);

+static struct memcg_kmem_cache *dcache_pick_cache(void)
+{
+   struct mem_cgroup *memcg;
+   struct memcg_kmem_cache *kmem = &dentry_cache;
+   +
+   rcu_read_lock();
+   memcg = mem_cgroup_from_task(current);
+   rcu_read_unlock();
+   +
+   if (memcg)
+   kmem = memcg_cache_get(memcg, CACHE_DENTRY);
+   +
+   return kmem;
+}
+
/**
 * __d_alloc - allocate a dcache entry
 * @sb: filesystem it will belong to
@@ -1192,15 +1208,18 @@ struct dentry *__d_alloc(struct super_block *sb, const struct qstr
 *name)
{
    struct dentry *dentry;
    char *dname;
+   struct memcg_kmem_cache *cache;
+   +
+   cache = dcache_pick_cache();

-   dentry = kmem_cache_alloc(dentry_cache, GFP_KERNEL);
+   dentry = kmem_cache_alloc(cache->cache, GFP_KERNEL);
    if (!dentry)
        return NULL;

    if (name->len > DNAME_INLINE_LEN-1) {
        dname = kmalloc(name->len + 1, GFP_KERNEL);
        if (!dname) {
-       kmem_cache_free(dentry_cache, dentry);

```

```

+ kmem_cache_free(cache->cache, dentry);
  return NULL;
}
} else {
@@ -1222,6 +1241,7 @@ struct dentry *__d_alloc(struct super_block *sb, const struct qstr
*name)
  dentry->d_sb = sb;
  dentry->d_op = NULL;
  dentry->d_fsdata = NULL;
+ dentry->d_cache = cache;
  INIT_HLIST_BL_NODE(&dentry->d_hash);
  INIT_LIST_HEAD(&dentry->d_lru);
  INIT_LIST_HEAD(&dentry->d_subdirs);
@@ -2990,6 +3010,11 @@ static void __init dcache_init_early(void)
  INIT_HLIST_BL_HEAD(dentry_hashtable + loop);
}

+struct memcg_cache_struct memcg_dcache = {
+ .index = CACHE_DENTRY,
+ .shrink_fn = dcache_shrink_memcg,
+};
+
static void __init dcache_init(void)
{
  int loop;
@@ -2999,7 +3024,7 @@ static void __init dcache_init(void)
  * but it is probably not worth it because of the cache nature
  * of the dcache.
  */
- dentry_cache = KMEM_CACHE(dentry,
+ dentry_cache.cache = KMEM_CACHE(dentry,
  SLAB_RECLAIM_ACCOUNT|SLAB_PANIC|SLAB_MEM_SPREAD);

  /* Hash may have been set up in dcache_init_early */
@@ -3018,6 +3043,9 @@ static void __init dcache_init(void)

  for (loop = 0; loop < (1 << d_hash_shift); loop++)
    INIT_HLIST_BL_HEAD(dentry_hashtable + loop);
+
+ memcg_dcache.cache = dentry_cache.cache;
+ register_memcg_cache(&memcg_dcache);
}

/* SLAB cache for __getname() consumers */
diff --git a/include/linux/dcache.h b/include/linux/dcache.h
index d64a55b..4d94b657 100644
--- a/include/linux/dcache.h
+++ b/include/linux/dcache.h

```

```

@@ -113,6 +113,8 @@ full_name_hash(const unsigned char *name, unsigned int len)
# endif
#endif

+struct mem_cgroup;
+
struct dentry {
    /* RCU lookup touched fields */
    unsigned int d_flags; /* protected by d_lock */
@@ -142,6 +144,7 @@ struct dentry {
    } d_u;
    struct list_head d_subdirs; /* our children */
    struct list_head d_alias; /* inode alias list */
+ struct memcg_kmem_cache *d_cache;
};

/*
--
1.7.7.6

```
