

---

Subject: [-mm PATCH 3/10] Memory controller accounting setup (v7)

Posted by [Balbir Singh](#) on Fri, 24 Aug 2007 15:20:19 GMT

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

---

From: Pavel Emelianov <xemul@openvz.org>

Changelog for v5

1. Remove inclusion of memcontrol.h from mm\_types.h

Changelog

As per Paul's review comments

1. Drop css\_get() for the root memory container
2. Use mem\_container\_from\_task() as an optimization instead of using mem\_container\_from\_cont() along with task\_container.

Basic setup routines, the mm\_struct has a pointer to the container that it belongs to and the page has a page\_container associated with it.

Signed-off-by: Pavel Emelianov <xemul@openvz.org>

Signed-off-by: Balbir Singh <balbir@linux.vnet.ibm.com>

---

```
include/linux/memcontrol.h | 36 ++++++
include/linux/mm_types.h   |  6 ++++
include/linux/sched.h      |  1
kernel/fork.c              | 11 ++++++
mm/memcontrol.c            | 57 ++++++
5 files changed, 104 insertions(+), 7 deletions(-)
```

```
diff -puN include/linux/memcontrol.h~mem-control-accounting-setup include/linux/memcontrol.h
--- linux-2.6.23-rc2-mm2/include/linux/memcontrol.h~mem-control-accounting-setup 2007-08-24
20:46:07.000000000 +0530
```

```
+++ linux-2.6.23-rc2-mm2-balbir/include/linux/memcontrol.h 2007-08-24 20:46:07.000000000
+0530
```

```
@ @ -3,6 +3,9 @ @
```

```
* Copyright IBM Corporation, 2007
```

```
* Author Balbir Singh <balbir@linux.vnet.ibm.com>
```

```
*
```

```
+ * Copyright 2007 OpenVZ SWsoft Inc
```

```
+ * Author: Pavel Emelianov <xemul@openvz.org>
```

```
+ *
```

```
* This program is free software; you can redistribute it and/or modify
```

```
* it under the terms of the GNU General Public License as published by
```

```
* the Free Software Foundation; either version 2 of the License, or
```

@@ -17,5 +20,38 @@

```
#ifndef _LINUX_MEMCONTROL_H
#define _LINUX_MEMCONTROL_H
```

```
+struct mem_container;
+struct page_container;
+
+#ifdef CONFIG_CONTAINER_MEM_CONT
+
+extern void mm_init_container(struct mm_struct *mm, struct task_struct *p);
+extern void mm_free_container(struct mm_struct *mm);
+extern void page_assign_page_container(struct page *page,
+    struct page_container *pc);
+extern struct page_container *page_get_page_container(struct page *page);
+
+#else /* CONFIG_CONTAINER_MEM_CONT */
+static inline void mm_init_container(struct mm_struct *mm,
+    struct task_struct *p)
+{
+}
+
+static inline void mm_free_container(struct mm_struct *mm)
+{
+}
+
+static inline void page_assign_page_container(struct page *page,
+    struct page_container *pc)
+{
+}
+
+static inline struct page_container *page_get_page_container(struct page *page)
+{
+    return NULL;
+}
+
+#endif /* CONFIG_CONTAINER_MEM_CONT */
+
+#endif /* _LINUX_MEMCONTROL_H */
```

```
diff -puN include/linux/mm_types.h~mem-control-accounting-setup include/linux/mm_types.h
--- linux-2.6.23-rc2-mm2/include/linux/mm_types.h~mem-control-accounting-setup 2007-08-24
20:46:07.000000000 +0530
+++ linux-2.6.23-rc2-mm2-balbir/include/linux/mm_types.h 2007-08-24 20:46:07.000000000
+0530
```

```
@@ -96,6 +96,9 @@ struct page {
    unsigned int gfp_mask;
    unsigned long trace[8];
#endif
```

```

#ifdef CONFIG_CONTAINER_MEM_CONT
+ unsigned long page_container;
#endif
};

/*
@@ -227,6 +230,9 @@ struct mm_struct {
/* aio bits */
rwlock_t ioctx_list_lock;
struct kiocx *ioctx_list;
#ifdef CONFIG_CONTAINER_MEM_CONT
+ struct mem_container *mem_container;
#endif
};

#endif /* _LINUX_MM_TYPES_H */
diff -puN include/linux/sched.h~mem-control-accounting-setup include/linux/sched.h
--- linux-2.6.23-rc2-mm2/include/linux/sched.h~mem-control-accounting-setup 2007-08-24
20:46:07.000000000 +0530
+++ linux-2.6.23-rc2-mm2-balbir/include/linux/sched.h 2007-08-24 20:46:07.000000000 +0530
@@ -88,6 +88,7 @@ struct sched_param {

#include <asm/processor.h>

+struct mem_container;
struct exec_domain;
struct futex_pi_state;
struct bio;
diff -puN kernel/fork.c~mem-control-accounting-setup kernel/fork.c
--- linux-2.6.23-rc2-mm2/kernel/fork.c~mem-control-accounting-setup 2007-08-24
20:46:07.000000000 +0530
+++ linux-2.6.23-rc2-mm2-balbir/kernel/fork.c 2007-08-24 20:46:07.000000000 +0530
@@ -51,6 +51,7 @@
#include <linux/random.h>
#include <linux/tty.h>
#include <linux/proc_fs.h>
#include <linux/memcontrol.h>

#include <asm/pgtable.h>
#include <asm/pgalloc.h>
@@ -329,7 +330,7 @@ __cacheline_aligned_in_smp DEFINE_SPINLO

#include <linux/init_task.h>

-static struct mm_struct * mm_init(struct mm_struct * mm)
+static struct mm_struct * mm_init(struct mm_struct * mm, struct task_struct *p)
{
atomic_set(&mm->mm_users, 1);

```

```

    atomic_set(&mm->mm_count, 1);
@@ -346,11 +347,14 @@ static struct mm_struct * mm_init(struct
    mm->iocx_list = NULL;
    mm->free_area_cache = TASK_UNMAPPED_BASE;
    mm->cached_hole_size = ~0UL;
+ mm_init_container(mm, p);

    if (likely(!mm_alloc_pgd(mm))) {
        mm->def_flags = 0;
        return mm;
    }
+
+ mm_free_container(mm);
    free_mm(mm);
    return NULL;
}
@@ -365,7 +369,7 @@ struct mm_struct * mm_alloc(void)
    mm = allocate_mm();
    if (mm) {
        memset(mm, 0, sizeof(*mm));
- mm = mm_init(mm);
+ mm = mm_init(mm, current);
    }
    return mm;
}
@@ -379,6 +383,7 @@ void fastcall __mmdrop(struct mm_struct
{
    BUG_ON(mm == &init_mm);
    mm_free_pgd(mm);
+ mm_free_container(mm);
    destroy_context(mm);
    free_mm(mm);
}
@@ -499,7 +504,7 @@ static struct mm_struct *dup_mm(struct t
    mm->token_priority = 0;
    mm->last_interval = 0;

- if (!mm_init(mm))
+ if (!mm_init(mm, tsk))
    goto fail_nomem;

    if (init_new_context(tsk, mm))
diff -puN mm/memcontrol.c~mem-control-accounting-setup mm/memcontrol.c
--- linux-2.6.23-rc2-mm2/mm/memcontrol.c~mem-control-accounting-setup 2007-08-24
20:46:07.000000000 +0530
+++ linux-2.6.23-rc2-mm2-balbir/mm/memcontrol.c 2007-08-24 20:46:07.000000000 +0530
@@ -3,6 +3,9 @@
 * Copyright IBM Corporation, 2007

```

```

* Author Balbir Singh <balbir@linux.vnet.ibm.com>
*
+ * Copyright 2007 OpenVZ SWsoft Inc
+ * Author: Pavel Emelianov <xemul@openvz.org>
+ *
* This program is free software; you can redistribute it and/or modify
* it under the terms of the GNU General Public License as published by
* the Free Software Foundation; either version 2 of the License, or
@@ -17,6 +20,7 @@
#include <linux/res_counter.h>
#include <linux/memcontrol.h>
#include <linux/container.h>
+#include <linux/mm.h>

struct container_subsys mem_container_subsys;

@@ -35,6 +39,13 @@ struct mem_container {
    * the counter to account for memory usage
    */
    struct res_counter res;
+ /*
+  * Per container active and inactive list, similar to the
+  * per zone LRU lists.
+  * TODO: Consider making these lists per zone
+  */
+ struct list_head active_list;
+ struct list_head inactive_list;
};

/*
@@ -56,6 +67,37 @@ struct mem_container *mem_container_from
    css);
}

+static inline
+struct mem_container *mem_container_from_task(struct task_struct *p)
+{
+ return container_of(task_subsys_state(p, mem_container_subsys_id),
+ struct mem_container, css);
+}
+
+void mm_init_container(struct mm_struct *mm, struct task_struct *p)
+{
+ struct mem_container *mem;
+
+ mem = mem_container_from_task(p);
+ css_get(&mem->css);
+ mm->mem_container = mem;

```

```

+}
+
+void mm_free_container(struct mm_struct *mm)
+{
+ css_put(&mm->mem_container->css);
+}
+
+void page_assign_page_container(struct page *page, struct page_container *pc)
+{
+ page->page_container = (unsigned long)pc;
+}
+
+struct page_container *page_get_page_container(struct page *page)
+{
+ return page->page_container;
+}
+
+static ssize_t mem_container_read(struct container *cont, struct cftype *cft,
+    struct file *file, char __user *userbuf, size_t nbytes,
+    loff_t *ppos)
@@ -91,14 +133,21 @@ static struct cftype mem_container_files
},
};

+static struct mem_container init_mem_container;
+
+static struct container_subsys_state *
+mem_container_create(struct container_subsys *ss, struct container *cont)
+{
+ struct mem_container *mem;

- mem = kzalloc(sizeof(struct mem_container), GFP_KERNEL);
- if (!mem)
- return -ENOMEM;
+ if (unlikely((cont->parent) == NULL)) {
+ mem = &init_mem_container;
+ init_mm.mem_container = mem;
+ } else
+ mem = kzalloc(sizeof(struct mem_container), GFP_KERNEL);
+
+ if (mem == NULL)
+ return NULL;

    res_counter_init(&mem->res);
    return &mem->css;
@@ -123,5 +172,5 @@ struct container_subsys mem_container_su
.create = mem_container_create,
.destroy = mem_container_destroy,

```

```
.populate = mem_container_populate,  
- .early_init = 0,  
+ .early_init = 1,  
};
```

—

--

Warm Regards,  
Balbir Singh  
Linux Technology Center  
IBM, ISTL

---

Containers mailing list  
Containers@lists.linux-foundation.org  
<https://lists.linux-foundation.org/mailman/listinfo/containers>

---