
Subject: [RFC][PATCH 7/7] Account for the number of files opened within container
Posted by [xemul](#) on Tue, 06 Mar 2007 15:05:46 GMT

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

Simple again - increment usage counter at file open and
decrement at file close. Reject opening if limit is hit.

```
diff -upr linux-2.6.20.orig/fs/Makefile linux-2.6.20-0/fs/Makefile
--- linux-2.6.20.orig/fs/Makefile 2007-02-04 21:44:54.000000000 +0300
+++ linux-2.6.20-0/fs/Makefile 2007-03-06 13:33:28.000000000 +0300
@@ -19,6 +19,8 @@ else
obj-y += no-block.o
endif

+obj-$(CONFIG_FILES_CONTAINER) += numfiles_container.o
+
obj-$(CONFIG_INOTIFY) += inotify.o
obj-$(CONFIG_INOTIFY_USER) += inotify_user.o
obj-$(CONFIG_EPOLL) += eventpoll.o
diff -upr linux-2.6.20.orig/fs/file_table.c linux-2.6.20-0/fs/file_table.c
--- linux-2.6.20.orig/fs/file_table.c 2007-02-04 21:44:54.000000000 +0300
+++ linux-2.6.20-0/fs/file_table.c 2007-03-06 13:33:28.000000000 +0300
@@ -21,6 +21,7 @@
#include <linux/fsnotify.h>
#include <linux/sysctl.h>
#include <linux/percpu_counter.h>
+#include <linux/numfiles_container.h>

#include <asm/atomic.h>

@@ -42,6 +43,7 @@ static inline void file_free_rcu(struct

static inline void file_free(struct file *f)
{
+ container_file_uncharge(f);
percpu_counter_dec(&nr_files);
call_rcu(&f->f_u.fu_rcuhead, file_free_rcu);
}
@@ -109,6 +111,10 @@ struct file *get_empty_filp(void)

percpu_counter_inc(&nr_files);
memset(f, 0, sizeof(*f));
+
+ if (container_file_charge(f))
+ goto fail_charge;
+
if (security_file_alloc(f))
goto fail_sec;
```

```
@@ -132,7 +138,10 @@ over:
    goto fail;
```

```
fail_sec:
- file_free(f);
+ container_file_uncharge(f);
+fail_charge:
+ percpu_counter_dec(&nr_files);
+ kmem_cache_free(filp_cachep, f);
fail:
    return NULL;
}
```

```
diff -upr linux-2.6.20.orig/fs/numfiles_container.c linux-2.6.20-0/fs/numfiles_container.c
--- linux-2.6.20.orig/fs/numfiles_container.c 2007-03-06 13:39:17.000000000 +0300
+++ linux-2.6.20-0/fs/numfiles_container.c 2007-03-06 13:33:28.000000000 +0300
```

```
@@ -0,0 +1,152 @@
```

```
+/*
```

```
+ * Numfiles accounting container
```

```
+ *
```

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

```
+ *
```

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

```
+ *
```

```
+ */
```

```
+
```

```
+#include <linux/list.h>
```

```
+#include <linux/sched.h>
```

```
+#include <linux/mm.h>
```

```
+#include <linux/res_counter.h>
```

```
+#include <linux/numfiles_container.h>
```

```
+
```

```
+static struct container_subsys numfiles_subsys;
```

```
+
```

```
+struct files_container {
```

```
+ struct res_counter res;
```

```
+ struct container_subsys_state css;
```

```
+};
```

```
+
```

```
+static inline struct files_container *numfiles_from_cont(struct container *cnt)
```

```
+{
```

```
+ return container_of(container_subsys_state(cnt, &numfiles_subsys),
```

```
+ struct files_container, css);
```

```
+}
```

```
+
```

```
+int container_file_charge(struct file *file)
```

```
+{
```

```
+ struct files_container *fc;
```

```

+
+ rcu_read_lock();
+ fc = numfiles_from_cont(task_container(current, &numfiles_subsys));
+ css_get_current(&fc->css);
+ rcu_read_unlock();
+
+
+ if (res_counter_charge(&fc->res, 1)) {
+  css_put(&fc->css);
+  return -ENOMEM;
+ }
+
+ file->f_cont = fc;
+ return 0;
+}
+
+void container_file_uncharge(struct file *file)
+{
+ struct files_container *fc;
+
+ fc = file->f_cont;
+ res_counter_uncharge(&fc->res, 1);
+ css_put(&fc->css);
+}
+
+static int numfiles_create(struct container_subsys *ss, struct container *cont)
+{
+ struct files_container *fc;
+
+ fc = kzalloc(sizeof(struct files_container), GFP_KERNEL);
+ if (fc == NULL)
+  return -ENOMEM;
+
+ res_counter_init(&fc->res);
+ cont->subsys[numfiles_subsys.subsys_id] = &fc->css;
+ return 0;
+}
+
+static void numfiles_destroy(struct container_subsys *ss,
+ struct container *cont)
+{
+ kfree(numfiles_from_cont(cont));
+}
+
+
+static ssize_t numfiles_read(struct container *cont, struct cftype *cft,
+ struct file *file, char __user *userbuf,
+ size_t nbytes, loff_t *ppos)
+{

```

```

+ return res_counter_read(&numfiles_from_cont(cont)->res, cft->private,
+ userbuf, nbytes, ppos);
+}
+
+static ssize_t numfiles_write(struct container *cont, struct cftype *cft,
+ struct file *file, const char __user *userbuf,
+ size_t nbytes, loff_t *ppos)
+{
+ return res_counter_write(&numfiles_from_cont(cont)->res, cft->private,
+ userbuf, nbytes, ppos);
+}
+
+
+static struct cftype numfiles_usage = {
+ .name = "numfiles_usage",
+ .private = RES_USAGE,
+ .read = numfiles_read,
+};
+
+static struct cftype numfiles_limit = {
+ .name = "numfiles_limit",
+ .private = RES_LIMIT,
+ .read = numfiles_read,
+ .write = numfiles_write,
+};
+
+static struct cftype numfiles_failcnt = {
+ .name = "numfiles_failcnt",
+ .private = RES_FAILCNT,
+ .read = numfiles_read,
+};
+
+static int numfiles_populate(struct container_subsys *ss,
+ struct container *cont)
+{
+ int rc;
+
+ if ((rc = container_add_file(cont, &numfiles_usage)) < 0)
+ return rc;
+ if ((rc = container_add_file(cont, &numfiles_failcnt)) < 0)
+ return rc;
+ if ((rc = container_add_file(cont, &numfiles_limit)) < 0)
+ return rc;
+
+ return 0;
+}
+
+static struct files_container init_files_container;

```

```

+
+static __init int numfiles_create_early(struct container_subsys *ss,
+ struct container *cont)
+{
+ struct files_container *np;
+
+ np = &init_files_container;
+ res_counter_init(&np->res);
+ cont->subsys[numfiles_subsys.subsys_id] = &np->css;
+ ss->create = numfiles_create;
+ return 0;
+}
+
+static struct container_subsys numfiles_subsys = {
+ .name = "numfiles",
+ .create = numfiles_create_early,
+ .destroy = numfiles_destroy,
+ .populate = numfiles_populate,
+};
+
+void __init container_numfiles_init_early(void)
+{
+ container_register_subsys(&numfiles_subsys);
+}
+
diff -upr linux-2.6.20.orig/include/linux/fs.h linux-2.6.20-0/include/linux/fs.h
--- linux-2.6.20.orig/include/linux/fs.h 2007-03-06 13:33:28.000000000 +0300
+++ linux-2.6.20-0/include/linux/fs.h 2007-03-06 13:33:28.000000000 +0300
@@ -739,6 +739,9 @@ struct file {
    spinlock_t f_ep_lock;
#ifdef CONFIG_EPOLL */
    struct address_space *f_mapping;
+#ifdef CONFIG_FILES_CONTAINER
+ struct files_container *f_cont;
+#endif
};
extern spinlock_t files_lock;
#define file_list_lock() spin_lock(&files_lock);
diff -upr linux-2.6.20.orig/include/linux/numfiles_container.h
linux-2.6.20-0/include/linux/numfiles_container.h
--- linux-2.6.20.orig/include/linux/numfiles_container.h 2007-03-06 13:39:17.000000000 +0300
+++ linux-2.6.20-0/include/linux/numfiles_container.h 2007-03-06 13:33:28.000000000 +0300
@@ -0,0 +1,33 @@
+#ifndef __NUMFILES_CONTAINER_H__
+#define __NUMFILES_CONTAINER_H__
+/*
+ * Numfiles container
+ *

```

```

+ * Copyright 2007 OpenVZ SWsoft Inc
+ *
+ * Author: Pavel Emelianov <xemul@openvz.org>
+ *
+ */
+
+#ifdef CONFIG_FILES_CONTAINER
+int container_file_charge(struct file *file);
+void container_file_uncharge(struct file *file);
+
+void container_numfiles_init_early(void);
+#else
+static inline int container_file_charge(struct file *file)
+{
+ return 0;
+}
+
+static inline void container_file_uncharge(struct file *file)
+{
+}
+
+static inline void container_numfiles_init_early(void)
+{
+}
+#endif
+
+#endif
+
diff -upr linux-2.6.20.orig/init/Kconfig linux-2.6.20-0/init/Kconfig
--- linux-2.6.20.orig/init/Kconfig 2007-03-06 13:33:28.000000000 +0300
+++ linux-2.6.20-0/init/Kconfig 2007-03-06 13:33:28.000000000 +0300
@@ -265,6 +265,12 @@ config CPUSETS
     help
         Provides the-number-of-tasks accounting container

+config FILES_CONTAINER
+ bool "Numfiles accounting container"
+ select RESOURCE_COUNTERS
+ help
+     Provides the-number-of-files accounting container
+
config SYSFS_DEPRECATED
 bool "Create deprecated sysfs files"
 default y
diff -upr linux-2.6.20.orig/kernel/container.c linux-2.6.20-0/kernel/container.c
--- linux-2.6.20.orig/kernel/container.c 2007-03-06 13:33:28.000000000 +0300
+++ linux-2.6.20-0/kernel/container.c 2007-03-06 13:35:48.000000000 +0300
@@ -60,6 +60,7 @@

```

```
#include <linux/rss_container.h>
#include <linux/numproc_container.h>
+#include <linux/numfiles_container.h>

#define CONTAINER_SUPER_MAGIC 0x27e0eb

@@ -1721,6 +1725,7 @@ int __init container_init_early(void)

    container_rss_init_early();
    container_numproc_init_early();
+ container_numfiles_init_early();

    return 0;
}
```
