
Subject: [PATCH 9/10] Bsdacct: stop accounting for all namespaces on umount time

Posted by [Pavel Emelianov](#) on Thu, 10 Apr 2008 08:37:14 GMT

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

All the `bsd_acct_strcts` with opened accounting are linked into a global list. So, the `acct_auto_close(_mnt)` walks one and drops the accounting for each.

BTW, it worth dropping the `acct->file != NULL` check from these routines, but I noticed this only when started sending the set :(... Will do it in the v2.

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

```
kernel/acct.c | 28 ++++++-----
1 files changed, 16 insertions(+), 12 deletions(-)
```

```
diff --git a/kernel/acct.c b/kernel/acct.c
index febbbc6..7fc9f9d 100644
```

```
--- a/kernel/acct.c
```

```
+++ b/kernel/acct.c
```

```
@@ -89,9 +89,11 @@ struct bsd_acct_struct {
    struct file *file;
    struct pid_namespace *ns;
    struct timer_list timer;
+ struct list_head list;
};
```

```
static DEFINE_SPINLOCK(acct_lock);
+static LIST_HEAD(acct_list);
```

```
/*
```

```
 * Called whenever the timer says to check the free space.
```

```
@@ -188,12 +190,14 @@ static void acct_file_reopen(struct bsd_acct_struct *acct, struct file
*file,
    acct->needcheck = 0;
    acct->file = NULL;
    acct->ns = NULL;
+ list_del(&acct->list);
}
if (file) {
    acct->file = file;
    acct->ns = ns;
    acct->needcheck = 0;
    acct->active = 1;
+ list_add(&acct->list, &acct_list);
```

```

/* It's been deleted if it was used before so this is safe */
setup_timer(&acct->timer, acct_timeout, (unsigned long)acct);
acct->timer.expires = jiffies + ACCT_TIMEOUT*HZ;
@@ -314,13 +318,13 @@ void acct_auto_close_mnt(struct vfsmount *m)
{
    struct bsd_acct_struct *acct;

- acct = init_pid_ns.bacct;
- if (acct == NULL)
- return;
-
    spin_lock(&acct_lock);
- if (acct->file && acct->file->f_path.mnt == m)
- acct_file_reopen(acct, NULL, NULL);
+restart:
+ list_for_each_entry(acct, &acct_list, list)
+ if (acct->file && acct->file->f_path.mnt == m) {
+ acct_file_reopen(acct, NULL, NULL);
+ goto restart;
+ }
    spin_unlock(&acct_lock);
}

```

```

@@ -335,13 +339,13 @@ void acct_auto_close(struct super_block *sb)
{
    struct bsd_acct_struct *acct;

- acct = init_pid_ns.bacct;
- if (acct == NULL)
- return;
-
    spin_lock(&acct_lock);
- if (acct->file && acct->file->f_path.mnt->mnt_sb == sb)
- acct_file_reopen(acct, NULL, NULL);
+restart:
+ list_for_each_entry(acct, &acct_list, list)
+ if (acct->file && acct->file->f_path.mnt->mnt_sb == sb) {
+ acct_file_reopen(acct, NULL, NULL);
+ goto restart;
+ }
    spin_unlock(&acct_lock);
}

```

--
1.5.3.4

Containers mailing list

