
Subject: [PATCH 1/3][UNIX] Make unix_tot_inflight counter non-atomic
Posted by [Pavel Emelianov](#) on Wed, 07 Nov 2007 13:52:36 GMT
[View Forum Message](#) <> [Reply to Message](#)

This counter is `_always_` modified under the `unix_gc_lock` spinlock,
so its atomicity can be provided w/o additional efforts.

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

diff --git a/include/net/af_unix.h b/include/net/af_unix.h
index 0864a77..a1c805d 100644

--- a/include/net/af_unix.h

+++ b/include/net/af_unix.h

@@ -12,7 +12,7 @@ extern void unix_gc(void);

#define UNIX_HASH_SIZE 256

-extern atomic_t unix_tot_inflight;

+extern unsigned int unix_tot_inflight;

struct unix_address {
 atomic_t refcnt;

diff --git a/net/unix/af_unix.c b/net/unix/af_unix.c
index 515e7a6..ab9048a 100644

--- a/net/unix/af_unix.c

+++ b/net/unix/af_unix.c

@@ -457,7 +457,7 @@ static int unix_release_sock (struct sock *sk, int embrion)

* What the above comment does talk about? --ANK(980817)

*/

- if (atomic_read(&unix_tot_inflight))

+ if (unix_tot_inflight)

unix_gc(); /* Garbage collect fds */

return 0;

diff --git a/net/unix/garbage.c b/net/unix/garbage.c
index 406b643..399717e 100644

--- a/net/unix/garbage.c

+++ b/net/unix/garbage.c

@@ -92,7 +92,7 @@ static LIST_HEAD(gc_inflight_list);

static LIST_HEAD(gc_candidates);

static DEFINE_SPINLOCK(unix_gc_lock);

-atomic_t unix_tot_inflight = ATOMIC_INIT(0);

+unsigned int unix_tot_inflight;

```

static struct sock *unix_get_socket(struct file *filp)
@@ -133,7 +133,7 @@ void unix_inflight(struct file *fp)
    } else {
        BUG_ON(list_empty(&u->link));
    }
- atomic_inc(&unix_tot_inflight);
+ unix_tot_inflight++;
    spin_unlock(&unix_gc_lock);
}
}
@@ -147,7 +147,7 @@ void unix_notinflight(struct file *fp)
    BUG_ON(list_empty(&u->link));
    if (atomic_dec_and_test(&u->inflight))
        list_del_init(&u->link);
- atomic_dec(&unix_tot_inflight);
+ unix_tot_inflight--;
    spin_unlock(&unix_gc_lock);
}
}
--
1.5.3.4

```
