Subject: [PATCH] Remove rcu_assign_pointer() penalty for NULL pointers
Posted by paulmck on Sat, 01 Dec 2007 00:37:21 GMT
View Forum Message <> Reply to Message

Hello!

The rcu_assign_pointer() primitive currently unconditionally executes
a memory barrier, even when a NULL pointer is being assigned. This
has lead some to avoid using rcu_assign_pointer() for NULL pointers,
which loses the self-documenting advantages of rcu_assign_pointer()
This patch uses __builtin_const_p() to omit needless memory barriers
for NULL-pointer assignments at compile time with no runtime penalty,
as discussed in the following thread:

http://www.mail-archive.com/netdev@vger.kernel.org/msg54852.html

Tested on x86_64 and ppc64, also compiled the four cases (NULL/non-NULL
and const/non-const) with gcc version 4.1.2, and hand-checked the
assembly output.

Signed-off-by: Paul E. McKenney <paulmck@linux.vnet.ibm.com>
---

rcupdate.h | 11 +++++++----
1 file changed, 7 insertions(+), 4 deletions(-)

diff -urpNa -X dontdiff linux-2.6.24-rc1-ego/include/linux/rcupdate.h
linux-2.6.24-rc1-egoxu/include/linux/rcupdate.h
--- linux-2.6.24-rc1-ego/include/linux/rcupdate.h 2007-11-06 15:30:02.000000000 -0800
+++ linux-2.6.24-rc1-egoxu/include/linux/rcupdate.h 2007-11-30 09:06:11.000000000 -0800
@@ -191,10 +191,13 @@ static inline void rcu_preempt_boost(voi
* code.
*/

-#define rcu_assign_pointer(p, v) ({ \
- smp_wmb(); \
- (p) = (v); \
- })
+#define rcu_assign_pointer(p, v) \
+ ({ \
+ if (!__builtin_constant_p(v) || \
+ ((v) != NULL)) \
+ smp_wmb(); \
+ (p) = (v); \
+ })

/**
* synchronize_sched - block until all CPUs have exited any non-preemptive